



Junos[®] OS

Security Configuration Guide

Release

10.3



Published: 2010-07-23

Revision 01

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986–1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Junos OS Security Configuration Guide
Release 10.3
Copyright © 2010, Juniper Networks, Inc.
All rights reserved. Printed in USA.

Revision History
July 2010—Revision 01

The information in this document is current as of the date listed in the revision history.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions. Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details. For complete product documentation, please see the Juniper Networks website at www.juniper.net/techpubs.

END USER LICENSE AGREEMENT

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

- a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.
- b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.
- c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.
- d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.
- e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14 (ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at <http://www.gnu.org/licenses/gpl.html>, and a copy of the LGPL at <http://www.gnu.org/licenses/lgpl.html>.

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

	About This Guide	xxxix
Part 1	Introduction to Junos OS	
Chapter 1	Introducing Junos OS for SRX Series Services Gateways	3
Chapter 2	Understanding IPv6 Flow-Based Processing	45
Chapter 3	Introducing Junos OS for J Series Services Routers	69
Part 2	Security Zones and Interfaces	
Chapter 4	Security Zones and Interfaces	85
Chapter 5	Address Books and Address Sets	103
Part 3	Security Policies	
Chapter 6	Security Policies	115
Chapter 7	Security Policy Schedulers	135
Chapter 8	Security Policy Applications	139
Part 4	Application Layer Gateways	
Chapter 9	ALGs	169
Chapter 10	H.323 ALGs	173
Chapter 11	ALG for IKE and ESP	199
Chapter 12	SIP ALGs	207
Chapter 13	SCCP ALGs	245
Chapter 14	MGCP ALGs	261
Chapter 15	RPC ALGs	287
Part 5	User Authentication	
Chapter 16	Firewall User Authentication	297
Chapter 17	Infranet Authentication	327
Part 6	Virtual Private Networks	
Chapter 18	Internet Protocol Security	355
Chapter 19	Public Key Cryptography for Certificates	383
Chapter 20	Dynamic VPNs	405
Chapter 21	Group VPNs	423

Part 7	Intrusion Detection and Prevention	
Chapter 22	IDP Policies	463
Chapter 23	Application-Level Distributed Denial of Service	523
Chapter 24	IDP Signature Database	535
Chapter 25	IDP Application Identification	549
Chapter 26	IDP SSL Inspection	563
Chapter 27	IDP Performance and Capacity Tuning	569
Chapter 28	IDP Logging	571
Part 8	Unified Threat Management	
Chapter 29	Unified Threat Management Overview	581
Chapter 30	Antispam Filtering	587
Chapter 31	Full Antivirus Protection	605
Chapter 32	Express Antivirus Protection	649
Chapter 33	Content Filtering	665
Chapter 34	Web Filtering	679
Part 9	Attack Detection and Prevention	
Chapter 35	Attack Detection and Prevention	709
Chapter 36	Reconnaissance Deterrence	711
Chapter 37	Suspicious Packet Attributes	735
Chapter 38	Denial-of-Service Attacks	743
Part 10	Application Identification	
Chapter 39	Junos OS Application Identification	769
Chapter 40	AppTrack Application Tracking	789
Part 11	Chassis Cluster	
Chapter 41	Chassis Cluster	795
Part 12	Network Address Translation	
Chapter 42	Network Address Translation	927
Part 13	GPRS	
Chapter 43	General Packet Radio Service	1013
Part 14	Index	
	Index	1033

Table of Contents

	About This Guide	xxxix
	J Series and SRX Series Documentation and Release Notes	xxxix
	Objectives	xl
	Audience	xl
	Supported Routing Platforms	xl
	Document Conventions	xl
	Documentation Feedback	xl
	Requesting Technical Support	xl
	Self-Help Online Tools and Resources	xl
	Opening a Case with JTAC	xl
Part 1	Introduction to Junos OS	
Chapter 1	Introducing Junos OS for SRX Series Services Gateways	3
	SRX Series Services Gateways Processing Overview	3
	Understanding Flow-Based Processing	4
	Zones and Policies	5
	Flows and Sessions	5
	Understanding Packet-Based Processing	5
	Stateless Firewall Filters	6
	Class-of-Service Features	6
	Screens	6
	Sessions for SRX Series Services Gateways	7
	Session Characteristics for SRX Series Services Gateways	7
	Understanding Session Characteristics for SRX Series Services Gateways	7
	Example: Controlling Session Termination for SRX Series Services Gateways	8
	Example: Disabling TCP Packet Security Checks for SRX Series Services Gateways	9
	Example: Setting the Maximum Segment Size for All TCP Sessions for SRX Series Services Gateways	10
	Monitoring Sessions for SRX Series Services Gateways	11
	Understanding How to Obtain Session Information for SRX Series Services Gateways	12
	Displaying Global Session Parameters for All SRX Series Services Gateways	12
	Displaying a Summary of Sessions for SRX Series Services Gateways	13
	Displaying Session and Flow Information About Sessions for SRX Series Services Gateways	14

Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways	14
Using Filters to Display Session and Flow Information for SRX Series Services Gateways	15
Information Provided in Session Log Entries for SRX Series Services Gateways	15
Clearing Sessions for SRX Series Services Gateways	18
Terminating Sessions for SRX Series Services Gateways	19
Terminating a Specific Session for SRX Series Services Gateways	19
Using Filters to Specify the Sessions to Be Terminated for SRX Series Services Gateways	19
Debugging for SRX Series Services Gateways	19
Data Path Debugging for SRX Series Services Gateways	19
Understanding Data Path Debugging for SRX Series Services Gateways	19
Debugging the Data Path (CLI Procedure)	20
Security Debugging for SRX Series Services Gateways	21
Understanding Security Debugging Using Trace Options	21
Setting Security Trace Options (CLI Procedure)	21
Displaying Output for Security Trace Options	22
Flow Debugging for SRX Series Services Gateways	22
Understanding Flow Debugging Using Trace Options	23
Setting Flow Debugging Trace Options (CLI Procedure)	23
Understanding SRX Series Services Gateways Central Point Architecture	23
Load Distribution in Combo Mode	24
Sharing Processing Power and Memory in Combo Mode	24
SRX5600 and SRX5800 Services Gateways Processing Overview	24
Understanding First-Packet Processing	25
Understanding Fast-Path Processing	27
Understanding the Data Path for Unicast Sessions	28
Session Lookup and Packet Match Criteria	28
Understanding Session Creation: First-Packet Processing	28
Understanding Fast-Path Processing	31
Understanding Packet Processing	34
Understanding Services Processing Units	35
Understanding Scheduler Characteristics	35
Understanding Network Processor Bundling	35
Network Processor Bundling Limitations	36
SRX3400 and SRX3600 Services Gateways Processing Overview	37
Components Involved in Setting up a Session	37
Understanding the Data Path for Unicast Sessions	38
Session Lookup and Packet Match Criteria	38
Understanding Session Creation: First Packet Processing	38
Understanding Fast-Path Processing	40
SRX210 Services Gateway Processing Overview	40
Understanding Flow Processing and Session Management	41
Understanding First-Packet Processing	41
Understanding Session Creation	41

	Understanding Fast-Path Processing	42
	Limitations of Flow and Processing	42
Chapter 2	Understanding IPv6 Flow-Based Processing	45
	Understanding IP Version 6 (IPv6)	46
	About the IPv6 Address Space, Addressing, and Address Types	46
	About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them	47
	About the IPv6 Address Format	48
	The IPv6 Packet Header and SRX Series and J-series Devices Overview	49
	About the IPv6 Basic Packet Header	50
	Understanding IPv6 Packet Header Extensions	52
	About IPv6 Packet Header Verification Performed by the Flow Module for SRX Series and J-series Devices	53
	Understanding How SRX Series and J-series Devices Handle ICMPv6 Packets	53
	Understanding Path MTU Messages for IPv6 Packets	55
	Understanding How SRX Series and J-series Devices Handle Packet Fragmentation for IPv6 Flows	57
	Understanding Sessions for IPv6 Flows	57
	Understanding SRX5600 and SRX5800 Architecture and Flow Processing	57
	Limitations of IPv6	60
	Enabling Flow-Based Processing for IPv6 Traffic	61
	Using Filters to Display IPv6 Session and Flow Information for SRX Series Services Gateways	63
Chapter 3	Introducing Junos OS for J Series Services Routers	69
	Understanding Stateful and Stateless Data Processing for J Series Services Routers	69
	Understanding Flow-Based Processing	70
	Zones and Policies	71
	Flows and Sessions	71
	Understanding Packet-Based Processing	71
	Stateless Firewall Filters	72
	Class-of-Service Features	73
	Session Characteristics for J Series Services Routers	73
	Understanding Session Characteristics for J Series Services Routers	73
	Example: Controlling Session Termination for J Series Services Routers	74
	Example: Disabling TCP Packet Security Checks for J Series Services Routers	76
	Example: Accommodating End-to-End TCP Communication for J Series Services Routers	77
	Understanding the Data Path for J Series Services Routers	79
	Understanding the Forwarding Processing	80
	Understanding the Session-Based Processing	80
	Session Lookup	80
	First-Packet Path Processing	81
	Fast-Path Processing	82
	Understanding Forwarding Features	82

Part 2**Security Zones and Interfaces****Chapter 4****Security Zones and Interfaces 85**

Security Zones and Interfaces Overview	85
Understanding Security Zone Interfaces	86
Understanding Interface Ports	86
Security Zones	86
Understanding Functional Zones	87
Understanding Security Zones	87
Example: Creating Security Zones	88
Host Inbound Traffic	90
Understanding How to Control Inbound Traffic Based on Traffic Types	90
Supported System Services for Host Inbound Traffic	91
Example: Controlling Inbound Traffic Based on Traffic Types	92
Protocols	95
Understanding How to Control Inbound Traffic Based on Protocols	95
Example: Controlling Inbound Traffic Based on Protocols	96
TCP-Reset Parameters	97
Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter	98
Example: Configuring the TCP-Reset Parameter	98
DNS	99
DNS Overview	99
DNS Components	99
DNS Server Caching	99
Forwarders	100
Example: Configuring the TTL Value for DNS name servers	100
Example: Configuring a Forwarder for a DNS server	100
DNSSEC Overview	100
Example: Configuring DNSSEC	101
Example: Configuring Keys for DNSSEC	101
Example: Configuring Secure Domains and Trusted Keys for DNSSEC	102

Chapter 5**Address Books and Address Sets 103**

Security Policy Address Books and Address Sets Overview	103
Understanding Address Books	104
Understanding Address Sets	105
Limitations of Addresses and Address Sets	107
Example: Configuring Address Books	108
Verifying Address Book Configuration	110

Part 3**Security Policies****Chapter 6****Security Policies 115**

Security Policies Overview	115
Understanding Security Policy Rules	118
Understanding Security Policy Elements	120
Security Policies Configuration Overview	120
Example: Configuring a Security Policy to Permit or Deny All Traffic	121
Example: Configuring a Security Policy to Permit or Deny Selected Traffic	125

	Understanding Security Policy Ordering	129
	Example: Reordering the Policies	130
	Troubleshooting Security Policies	131
	Checking a Security Policy Commit Failure	131
	Verifying a Security Policy Commit	132
	Debugging Policy Lookup	132
	Monitoring Policy Statistics	132
	Matching Security Policies	133
Chapter 7	Security Policy Schedulers	135
	Security Policy Schedulers Overview	135
	Example: Configuring Schedulers (CLI)	136
	Example: Associating a Policy to a Scheduler (CLI)	137
	Verifying Scheduled Policies	137
Chapter 8	Security Policy Applications	139
	Security Policy Applications Overview	139
	Policy Application Sets Overview	140
	Example: Configuring Applications and Application Sets	141
	Custom Policy Applications	142
	Understanding Custom Policy Applications	142
	Custom Application Mappings	142
	Example: Adding and Modifying Custom Policy Applications	143
	Example: Defining a Custom ICMP Application	144
	Policy Application Timeouts	146
	Understanding Policy Application Timeout Configuration and Lookup	146
	Understanding Policy Application Timeouts Contingencies	148
	Example: Setting a Policy Application Timeout	149
	Understanding the ICMP Predefined Policy Application	150
	Default Behaviour of ICMP Unreachable Errors	154
	Understanding Internet-Related Predefined Policy Applications	155
	Understanding Microsoft Predefined Policy Applications	156
	Understanding Dynamic Routing Protocols Predefined Policy Applications	157
	Understanding Streaming Video Predefined Policy Applications	158
	Understanding Sun RPC Predefined Policy Applications	159
	Understanding Security and Tunnel Predefined Policy Applications	160
	Understanding IP-Related Predefined Policy Applications	160
	Understanding Instant Messaging Predefined Policy Applications	161
	Understanding Management Predefined Policy Applications	162
	Understanding Mail Predefined Policy Applications	163
	Understanding UNIX Predefined Policy Applications	164
	Understanding Miscellaneous Predefined Policy Applications	164
Part 4	Application Layer Gateways	
Chapter 9	ALGs	169
	ALG Overview	169
	Understanding ALG Types	170

Chapter 10	H.323 ALGs	173
	Understanding H.323 ALGs	173
	Understanding the Avaya H.323 ALG	175
	Avaya H.323 ALG-Specific Features	175
	Call Flow Details in the Avaya H.323 ALG	175
	H.323 ALG Configuration Overview	176
	H.323 ALG Endpoint Registration Timeouts	177
	Understanding H.323 ALG Endpoint Registration Timeouts	177
	Example: Setting H.323 ALG Endpoint Registration Timeouts	177
	H.323 ALG Media Source Port Ranges	179
	Understanding H.323 ALG Media Source Port Ranges	179
	Example: Setting H.323 ALG Media Source Port Ranges	179
	H.323 ALG DoS Attack Protection	180
	Understanding H.323 ALG DoS Attack Protection	180
	Example: Configuring H.323 ALG DoS Attack Protection	181
	H.323 ALG Unknown Message Types	182
	Understanding H.323 ALG Unknown Message Types	182
	Example: Allowing Unknown H.323 ALG Message Types	183
	Example: Passing H.323 ALG Traffic to a Gatekeeper in the Internal Zone	184
	Example: Passing H.323 ALG Traffic to a Gatekeeper in the External Zone	189
	Example: Using NAT and the H.323 ALG to Enable Incoming Calls (CLI)	194
	Example: Using NAT and the H.323 ALG to Enable Outgoing Calls (CLI)	196
Chapter 11	ALG for IKE and ESP	199
	Understanding ALG for IKE and ESP	199
	Understanding ALG for IKE and ESP Operation	200
	Example: Configuring the IKE and ESP ALG (CLI)	200
	Example: Enabling IKE and ESP ALG and Setting Timeouts (CLI)	204
Chapter 12	SIP ALGs	207
	Understanding SIP ALGs	207
	SIP ALG Operation	208
	SDP Session Descriptions	209
	Pinhole Creation	210
	Understanding SIP ALG Request Methods	212
	SIP ALG Configuration Overview	213
	SIP ALG Call Duration and Timeouts	213
	Understanding SIP ALG Call Duration and Timeouts	213
	Example: Setting SIP ALG Call Duration and Timeouts	214
	SIP ALG DoS Attack Protection	216
	Understanding SIP ALG DoS Attack Protection	216
	Example: Configuring SIP ALG DoS Attack Protection	216
	SIP ALG Unknown Message Types	217
	Understanding SIP ALG Unknown Message Types	217
	Example: Allowing Unknown SIP ALG Message Types	218
	SIP ALG Hold Resources	219
	Understanding SIP ALG Hold Resources	219
	Retaining SIP ALG Hold Resources (J-Web Procedure)	220
	Retaining SIP ALG Hold Resources (CLI Procedure)	220

	SIP ALGs and NAT	220
	Understanding SIP ALGs and NAT	221
	Outgoing Calls	222
	Incoming Calls	222
	Forwarded Calls	223
	Call Termination	223
	Call Re-INVITE Messages	223
	Call Session Timers	223
	Call Cancellation	223
	Forking	224
	SIP Messages	224
	SIP Headers	224
	SIP Body	226
	SIP NAT Scenario	226
	Classes of SIP Responses	228
	Understanding Incoming SIP ALG Call Support Using the SIP Registrar and NAT	229
	Example: Configuring Interface Source NAT for Incoming SIP Calls (CLI)	230
	Example: Configuring a Source NAT Pool for Incoming SIP Calls (CLI)	232
	Example: Configuring Static NAT for Incoming SIP Calls (CLI)	234
	Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone (CLI)	235
	Example: Configuring the SIP Proxy and NAT in the Public Zone (CLI)	237
	Example: Configuring a Three-Zone SIP ALG and NAT Scenario (CLI)	238
	Verifying SIP ALG Configurations	241
	Verifying SIP ALGs	241
	Verifying SIP ALG Calls	241
	Verifying SIP ALG Call Details	242
	Verifying SIP ALG Counters	242
	Verifying the Rate of SIP ALG Messages	243
Chapter 13	SCCP ALGs	245
	Understanding SCCP ALGs	245
	SCCP Security	246
	SCCP Components	247
	SCCP Client	247
	CallManager	247
	Cluster	247
	SCCP Transactions	247
	Client Initialization	248
	Client Registration	248
	Call Setup	248
	Media Setup	248
	SCCP Control Messages and RTP Flow	248

	SCCP Messages	249
	SCCP ALG Configuration Overview	250
	SCCP ALG Inactive Media Timeout	251
	Understanding SCCP ALG Inactive Media Timeouts	251
	Example: Setting SCCP ALG Inactive Media Timeouts	251
	SCCP ALG Unknown Message Types	252
	Understanding SCCP ALG Unknown Message Types	252
	Example: Allowing Unknown SCCP ALG Message Types	253
	SCCP ALG DoS Attack Protection	254
	Understanding SCCP ALG DoS Attack Protection	254
	Example: Configuring SCCP ALG DoS Attack Protection	255
	Example: Configuring the SCCP ALG CallManager/TFTP Server in the Private Zone (CLI)	256
	Verifying SCCP ALG Configurations	257
	Verifying SCCP ALGs	257
	Verifying SCCP Calls	258
	Verifying SCCP Call Details	258
	Verifying SCCP Counters	259
Chapter 14	MGCP ALGs	261
	Understanding MGCP ALGs	261
	MGCP Security	262
	Entities in MGCP	262
	Endpoint	262
	Connection	263
	Call	263
	Call Agent	263
	Commands	264
	Response Codes	266
	MGCP ALG Configuration Overview	267
	MGCP ALG Call Duration and Timeouts	267
	Understanding MGCP ALG Call Duration and Timeouts	267
	Example: Setting MGCP ALG Call Duration	268
	Example: Setting MGCP ALG Inactive Media Timeout	270
	Example: Setting MGCP ALG Transaction Timeout	271
	MGCP ALG DoS Attack Protection	272
	Understanding MGCP ALG DoS Attack Protection	272
	Example: Configuring MGCP ALG DoS Attack Protection	272
	MGCP ALG Unknown Message Types	274
	Understanding MGCP ALG Unknown Message Types	274
	Example: Allowing Unknown MGCP ALG Message Types	274
	Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs	275
	Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALGs and NAT (CLI)	282

Chapter 15	RPC ALGs	287
	Understanding RPC ALGs	287
	Sun RPC ALGs	288
	Understanding Sun RPC ALGs	288
	Enabling Sun RPC ALGs (J-Web Procedure)	289
	Enabling Sun RPC ALGs (CLI Procedure)	289
	Sun RPC Services and Applications	289
	Understanding Sun RPC Services	290
	Customizing Sun RPC Applications (CLI Procedure)	290
	Microsoft RPC ALGs	291
	Understanding Microsoft RPC ALGs	291
	Enabling Microsoft RPC ALGs (J-Web Procedure)	292
	Enabling Microsoft RPC ALGs (CLI Procedure)	292
	Microsoft RPC Services and Applications	293
	Understanding Microsoft RPC Services	293
	Customizing Microsoft RPC Applications (CLI Procedure)	293
	Verifying the Microsoft RPC ALG Tables	293
Part 5	User Authentication	
Chapter 16	Firewall User Authentication	297
	Firewall User Authentication Overview	297
	Pass-Through Authentication	298
	Understanding Pass-Through Authentication	298
	Example: Configuring Pass-Through Authentication	299
	Web Authentication	304
	Understanding Web Authentication	305
	Example: Configuring Web Authentication	306
	External Authentication	312
	Understanding External Authentication Servers	312
	Understanding SecurID User Authentication	313
	Example: Configuring RADIUS and LDAP User Authentication	314
	Example: Configuring SecurID User Authentication	317
	Example: Deleting the SecurID Node Secret File	320
	Client Groups for Firewall Authentication	321
	Understanding Client Groups for Firewall Authentication	321
	Example: Configuring Local Users for Client Groups	322
	Firewall Authentication Banner Customization	323
	Understanding Firewall Authentication Banner Customization	323
	Example: Customizing a Firewall Authentication Banner	324

Chapter 17	Infranet Authentication	327
	UAC and Junos OS	327
	Understanding UAC in a Junos OS Environment	327
	Enabling UAC in a Junos OS Environment (CLI Procedure)	328
	Junos OS Enforcer and Infranet Controller Communications	329
	Understanding Communications Between the Junos OS Enforcer and the Infranet Controller	329
	Configuring Communications Between the Junos OS Enforcer and the Infranet Controller (CLI Procedure)	330
	Junos OS Enforcer Policy Enforcement	332
	Understanding Junos OS Enforcer Policy Enforcement	332
	Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure)	333
	Verifying Junos OS Enforcer Policy Enforcement	334
	Displaying Infranet Controller Authentication Table Entries from the Junos OS Enforcer	334
	Displaying Infranet Controller Resource Access Policies from the Junos OS Enforcer	334
	Junos OS Enforcer and IPsec	334
	Understanding Junos OS Enforcer Implementations Using IPsec	334
	Example: Configuring the Device as a Junos OS Enforcer Using IPsec (CLI)	336
	Junos OS Enforcer and Infranet Agent Endpoint Security	342
	Understanding Endpoint Security Using the Infranet Agent with the Junos OS Enforcer	342
	Configuring Endpoint Security Using the Infranet Agent with the Junos OS Enforcer	343
	Junos OS Enforcer and Captive Portal	343
	Understanding the Captive Portal on the Junos OS Enforcer	344
	Understanding Captive Portal Configuration on the Junos OS Enforcer	345
	Example: Creating a Captive Portal Policy on the Junos OS Enforcer (CLI)	346
	Understanding the Captive Portal Redirect URL Options	348
	Example: Configuring a Redirect URL for Captive Portal (CLI)	349
	Junos OS Enforcer and Infranet Controller Cluster Failover	351
	Understanding Communications Between Junos OS Enforcer and a Cluster of Infranet Controllers	351
	Configuring Junos OS Enforcer Failover Options (CLI Procedure)	351

Part 6 **Virtual Private Networks**

Chapter 18	Internet Protocol Security	355
	VPN Overview	355
	Security Associations	356
	IPsec Key Management	357
	Manual Key	357
	AutoKey IKE	357
	Diffie-Hellman Exchange	358

IPsec Security Protocols	358
AH Protocol	359
ESP Protocol	359
IPsec Tunnel Negotiation	360
Distributed VPNs in SRX Series Services Gateways	360
Understanding IKE and IPsec Packet Processing	361
Packet Processing in Tunnel Mode	361
IKE Packet Processing	362
IPsec Packet Processing	364
IPsec VPN Configuration Overview	366
Phase 1 Proposals for IPsec VPNs	367
Understanding Phase 1 of IKE Tunnel Negotiation	367
Main Mode	368
Aggressive Mode	368
Example: Configuring an IKE Phase 1 Proposal (CLI)	369
Example: Configuring an IKE Policy (CLI)	370
Example: Configuring an IKE Gateway (CLI)	370
Phase 2 Proposals for IPsec VPNs	371
Understanding Phase 2 of IKE Tunnel Negotiation	371
Proxy IDs	372
Perfect Forward Secrecy	372
Replay Protection	372
Example: Configuring an IPsec Phase 2 Proposal (CLI)	373
Example: Configuring an IPsec Policy (CLI)	373
Example: Configuring AutoKey IKE (CLI)	374
Global SPI and VPN Monitoring Features	374
Understanding Global SPI and VPN Monitoring Features	374
Example: Configuring Global SPI and VPN Monitoring Features (CLI)	375
Hub-and-Spoke VPNs	375
Understanding Hub-and-Spoke VPNs	375
Hub-and-Spoke VPN Configuration Overview	376
Example: Configuring the Hub in a Hub-and-Spoke VPN (CLI)	377
Example: Configuring Spoke 1 in a Hub-and-Spoke VPN (CLI)	380
Example: Configuring Spoke 2 in a Hub-and-Spoke VPN (CLI)	381
Chapter 19 Public Key Cryptography for Certificates	383
Understanding Public Key Infrastructure	383
PKI Hierarchy for a Single CA Domain or Across Domains	383
PKI Management and Implementation	385
Certificates and Certificate Authority	385
Understanding Certificates	386
Certificate Signatures	386
Certificate Verification	386
Internet Key Exchange	387
Digital Certificates Configuration Overview	387
Enabling Digital Certificates Online: Configuration Overview	388
Manually Generating Digital Certificates: Configuration Overview	388
Verifying the Validity of a Certificate: Configuration Overview	389

Deleting a Certificate: Configuration Overview	389
Public-Private Key Pairs	389
Understanding Public Key Cryptography	389
Example: Generating a Public-Private Key Pair (CLI)	390
Certificate Authority Profiles	390
Understanding Certificate Authority Profiles	390
Example: Configuring a Certificate Authority Profile (CLI)	391
Certificate Enrollment	391
Understanding Online CA Certificate Enrollment	391
Enrolling a CA Certificate Online (CLI Procedure)	392
Example: Enrolling a Local Certificate Online (CLI)	392
Example: Generating a Local Certificate Request Manually (CLI)	394
Example: Loading CA and Local Certificates Manually (CLI)	395
Example: Reenrolling Local Certificates Automatically (CLI)	396
Deleting Certificates (CLI Procedure)	397
Self-Signed Certificates	398
Understanding Self-Signed Certificates	398
Generating Self-Signed Certificates	398
Automatically Generating Self-Signed Certificates	399
Manually Generating Self-Signed Certificates	399
Using Automatically Generated Self-Signed Certificates (CLI Procedure)	399
Example: Manually Generating Self-Signed Certificates (CLI)	400
Certificate Revocation Lists	400
Understanding Certificate Revocation Lists	401
Example: Manually Loading a CRL onto the Device (CLI)	401
Example: Verifying Certificate Validity (CLI)	402
Example: Checking Certificate Validity Using CRLs (CLI)	403
Deleting a Loaded CRL (CLI Procedure)	403
Chapter 20 Dynamic VPNs	405
Dynamic VPN Overview	405
Dynamic VPN Configuration Overview	407
Dynamic VPN Client Configurations	408
Understanding Dynamic VPN Client Configurations	408
Example: Creating a Dynamic VPN Client Configuration (CLI)	409
Dynamic VPN Global Client Download Settings	409
Understanding Dynamic VPN Global Client Download Settings	409
Example: Configuring Dynamic VPN Global Client Download Settings (CLI)	410
Dynamic VPN and Access Manager User Experience	410
Understanding the Dynamic VPN and Access Manager User Experience	410
Connecting to the Remote Access Server for the First Time (Pre-IKE Phase)	411
Connecting to the Remote Access Server for Subsequent Sessions (Pre-IKE Phase)	412
Establishing an IPsec VPN Tunnel (IKE Phase)	413

	Access Manager Client-Side Reference	414
	Access Manager Client-Side System Requirements	414
	Access Manager Client-Side Files	415
	Access Manager Client-Side Registry Changes	417
	Access Manager Client-Side Error Messages	418
	Troubleshooting Access Manager Client-Side Problems	421
Chapter 21	Group VPNs	423
	Group VPN Overview	423
	Understanding the GDOI Protocol	425
	Understanding Group Servers and Members	426
	Understanding Dynamic Policies	427
	Group Key Operations	428
	Understanding Group Keys	428
	Understanding Rekey Messages	429
	Types of Rekey Messages	429
	Rekey Intervals	430
	Understanding Member Reregistration	430
	Understanding Key Activation	431
	Group VPN Configuration Overview	431
	Example: Configuring Group VPN (CLI)	432
	Overview	432
	Configuring the Group Server	433
	Configuring Member1	436
	Configuring Member2	438
	Viewing Dynamic Policies	442
	Understanding Colocation Mode	444
	Example: Configuring Group VPN with Server-Member Colocation (CLI)	445
	Understanding IKE Phase 1 Configuration for Group VPN	450
	Understanding IPsec SA Configuration for Group VPN	451
	Understanding VPN Group Configuration	452
	Understanding Antireplay	453
	Understanding Server-Member Communication	453
	Example: Configuring Server-Member Communication for Unicast Rekey Messages	454
	Example: Configuring Server-Member Communication for Multicast Rekey Messages	456
	Understanding Heartbeat Messages	457
	Understanding Group VPN Limitations	458
	Understanding Interoperability with Cisco GET VPN	458

Part 7

Intrusion Detection and Prevention

Chapter 22

IDP Policies	463
IDP Policies Overview	463
IDP Policy Terms	464
Working with IDP Policies	464
Example: Enabling IDP in a Security Policy	465
IDP Inline Tap Mode	468
Understanding IDP Inline Tap Mode	468
Example: Configuring IDP Inline Tap Mode	469
IDP Rules and Rulebases	470
Understanding IDP Policy Rules	470
Understanding IDP Rule Match Conditions	471
Understanding IDP Rule Objects	471
Understanding IDP Rule Actions	473
Understanding IDP Rule IP Actions	475
Understanding IDP Rule Notifications	475
IDP Rulebases	476
Understanding IDP Policy Rulebases	476
Example: Inserting a Rule in the IDP Rulebase	477
Example: Deactivating and Reactivating Rules in a IDP Rulebase	478
Understanding IDP Application-Level DDoS Rulebases	479
IDP IPS Rulebase	480
Understanding IDP IPS Rulebases	480
Example: Defining Rules for an IDP IPS Rulebase	481
IDP Exempt Rulebase	484
Understanding IDP Exempt Rulebases	484
Example: Defining Rules for an IDP Exempt Rulebase	485
IDP Terminal Rules	487
Understanding IDP Terminal Rules	487
Example: Setting Terminal Rules in Rulebases	488
IDP DSCP Rules	490
Understanding DSCP Rules in IDP Policies	490
Example: Configuring DSCP Rules in an IDP Policy	490
IDP Applications and Application Sets	493
Understanding IDP Application Sets	493
Example: Configuring IDP Applications and Services (CLI)	494
Example: Configuring IDP Applications Sets (CLI)	495
IDP Attacks and Attack Objects	496
Understanding Custom Attack Objects	496
Attack Name	496
Severity	496
Service and Application Bindings	497
Protocol and Port Bindings	500
Time Bindings	502
Attack Properties (Signature Attacks)	503
Attack Properties (Protocol Anomaly Attacks)	508

	Attack Properties (Compound or Chain Attacks)	509
	IDP Protocol Decoders	512
	Understanding IDP Protocol Decoders	512
	Example: Configuring IDP Protocol Decoders (CLI)	513
	Understanding Multiple IDP Detector Support	513
	IDP Signature-Based Attacks	514
	Understanding IDP Signature-Based Attacks	514
	Example: Configuring IDP Signature-Based Attacks (CLI)	515
	IDP Protocol Anomaly-Based Attacks	517
	Understanding IDP Protocol Anomaly-Based Attacks	517
	Example: Configuring IDP Protocol Anomaly-Based Attacks (CLI)	517
	Example: Specifying IDP Test Conditions for a Specific Protocol (CLI)	519
	Limitations of IDP	519
Chapter 23	Application-Level Distributed Denial of Service	523
	IDP Application-Level DDoS Attack Overview	523
	IDP Application-Level DDoS Protection Overview	523
	Understanding the Application-Level DDoS Module	524
	Understanding the Application-Level DDoS Definition	525
	Understanding the Application-Level DDoS Rule	526
	Understanding Application-Level DDoS IP-Action	527
	Understanding Application-Level DDoS Session Action	528
	Example: Enabling IDP Protection Against Application-Level DDoS Attacks (CLI)	528
	Understanding Application-level DDoS Statistic Reporting	531
	Example: Configuring Application-level DDoS Statistic Reporting	533
Chapter 24	IDP Signature Database	535
	Understanding the IDP Signature Database	535
	Example: Adding a Detector Sensor Configuration (J-Web)	536
	Predefined IDP Policy Templates	537
	Understanding Predefined IDP Policy Templates	537
	Downloading and Using Predefined IDP Policy Templates (CLI Procedure)	538
	IDP Signature Databases	539
	Understanding Predefined IDP Attack Objects and Object Groups	540
	Predefined Attack Objects	540
	Predefined Attack Object Groups	540
	Understanding the IDP Signature Database Version	541
	Updating the IDP Signature Database Overview	542
	Updating the IDP Signature Database Manually Overview	543
	Example: Updating the IDP Signature Database Manually (CLI)	543
	Example: Updating the Signature Database Automatically (CLI)	545
	Verifying the Signature Database	545
	Verifying the IDP Policy Compilation and Load Status	546
	Verifying the IDP Signature Database Version	547

Chapter 25	IDP Application Identification	549
	Understanding IDP Application Identification	549
	Understanding IDP Service and Application Bindings by Attack Objects	550
	Example: Configuring IDP Policies for Application Identification (CLI)	551
	Disabling Application Identification for an IDP Policy (CLI Procedure)	552
	IDP Application Identification for Nested Applications	553
	Understanding IDP Application Identification for Nested Applications	553
	Activating IDP Application Identification for Nested Applications (CLI Procedure)	554
	Example: Adding IDP Application Information to Attack Logging for Nested Applications (CLI)	554
	IDP Application System Cache	554
	Understanding the IDP Application System Cache	555
	Understanding IDP Application System Cache Information for Nested Application Identification	555
	Deactivating IDP Application System Cache Information for Nested Application Identification (CLI Procedure)	556
	Verifying Application System Cache Statistics	556
	IDP Memory and Session Limits	557
	Understanding Memory and Session Limit Settings for IDP Application Identification	557
	Example: Setting Memory and Session Limits for IDP Application Identification (CLI)	558
	Verifying IDP Counters for Application Identification Processes	559
Chapter 26	IDP SSL Inspection	563
	IDP SSL Overview	563
	Supported IDP SSL Ciphers	564
	Understanding IDP Internet Key Exchange	565
	Understanding IDP SSL Server Key Management and Policy Configuration	566
	Displaying IDP SSL Keys and Associated Servers	566
	Adding IDP SSL Keys and Associated Servers	567
	Deleting IDP SSL Keys and Associated Servers	567
	Configuring an IDP SSL Inspection (CLI Procedure)	568
Chapter 27	IDP Performance and Capacity Tuning	569
	Performance and Capacity Tuning for IDP Overview	569
	Configuring Session Capacity for IDP (CLI Procedure)	570
Chapter 28	IDP Logging	571
	Understanding IDP Logging	571
	Understanding Application-Level DDoS Logging	572
	Enabling Attack and IP-Action Logging (CLI Procedure)	573
	IDP Log Suppression Attributes	574
	Understanding IDP Log Suppression Attributes	574
	Example: Configuring IDP Log Suppression Attributes	575

	Understanding IDP Log Information Usage on the Infranet Controller	576
	Message Filtering to the Infranet Controller	576
	Configuring Infranet Controller Logging	576
	Security Packet Capture	577
	Understanding Security Packet Capture	577
	Example: Configuring Security Packet Capture (CLI)	577
	Example: Verifying Security Packet Capture (CLI)	578
Part 8	Unified Threat Management	
Chapter 29	Unified Threat Management Overview	581
	Unified Threat Management Overview	581
	Understanding UTM Custom Objects	582
	UTM Licensing	582
	Understanding UTM Licensing	583
	Updating UTM Licenses (CLI Procedure)	583
	WELF Logging for UTM Features	583
	Understanding WELF Logging for UTM Features	583
	Example: Configuring WELF Logging for UTM Features	584
Chapter 30	Antispam Filtering	587
	Antispam Filtering Overview	587
	Server-Based Spam Filtering	587
	Understanding Server-Based Antispam Filtering	587
	Server-Based Antispam Filtering Configuration Overview	588
	Example: Configuring Server-Based Antispam Filtering	589
	Local List Spam Filtering	594
	Understanding Local List Antispam Filtering	595
	Local List Antispam Filtering Configuration Overview	595
	Example: Configuring Local List Antispam Filtering	596
	Understanding Spam Message Handling	603
	Blocking Detected Spam	603
	Tagging Detected Spam	603
Chapter 31	Full Antivirus Protection	605
	Full Antivirus Protection Overview	605
	Full Antivirus Scanner Pattern Database	606
	Understanding Full Antivirus Pattern Updates	606
	Full Antivirus Pattern Update Configuration Overview	607
	Example: Specifying the Full Antivirus Pattern Update Server (CLI)	607
	Example: Automatically Updating Full Antivirus Patterns (J-Web)	608

Example: Automatically Updating Full Antivirus Patterns (CLI)	608
Manually Updating, Reloading, and Deleting Full Antivirus Patterns (CLI Procedure)	608
Full Antivirus File Scanning	609
Understanding the Full Antivirus Internal Scan Engine	609
Global, Profile-Based, and Policy-Based Full Antivirus Scan Settings	609
Understanding Full Antivirus Scan Level Settings	610
Example: Configuring Full Antivirus Scan Settings at Different Levels (CLI)	610
Full Antivirus Scan Modes	611
Understanding Full Antivirus Scan Mode Support	611
Configuring Full Antivirus File Extension Scanning (CLI Procedure)	611
Full Antivirus Intelligent Prescreening	612
Understanding Full Antivirus Intelligent Prescreening	612
Example: Configuring Full Antivirus Intelligent Prescreening (CLI)	612
Full Antivirus Content Size Limits	612
Understanding Full Antivirus Content Size Limits	613
Configuring Full Antivirus Content Size Limits (CLI Procedure)	613
Full Antivirus Decompression Layer Limit	613
Understanding Full Antivirus Decompression Layer Limits	613
Configuring Full Antivirus Decompression Layer Limits (CLI Procedure)	614
Full Antivirus Scanning Timeout	614
Understanding Full Antivirus Scanning Timeouts	614
Configuring Full Antivirus Scanning Timeouts (CLI Procedure)	615
Full Antivirus Scan Session Throttling	615
Understanding Full Antivirus Scan Session Throttling	615
Configuring Full Antivirus Scan Session Throttling (CLI Procedure)	615
Full Antivirus Application Protocol Scanning	615
Understanding Full Antivirus Application Protocol Scanning	616
HTTP Full Antivirus Scanning	617
Understanding HTTP Scanning	617
Enabling HTTP Scanning (CLI Procedure)	618
Understanding HTTP Trickling	618
Configuring HTTP Trickling to Prevent Timeouts During Antivirus Scanning (CLI Procedure)	618
Understanding MIME Whitelists	618
Example: Configuring MIME Whitelists to Bypass Antivirus Scanning (CLI)	619
Understanding URL Whitelists	619
Configuring URL Whitelists to Bypass Antivirus Scanning (CLI Procedure)	620
FTP Full Antivirus Scanning	620
Understanding FTP Antivirus Scanning	620
Enabling FTP Antivirus Scanning (CLI Procedure)	620
SMTP Full Antivirus Scanning	621
Understanding SMTP Antivirus Scanning	621
Enabling SMTP Antivirus Scanning (CLI Procedure)	622

POP3 Full Antivirus Scanning	622
Understanding POP3 Antivirus Scanning	623
Enabling POP3 Antivirus Scanning (CLI Procedure)	624
IMAP Full Antivirus Scanning	624
Understanding IMAP Antivirus Scanning	624
Enabling IMAP Antivirus Scanning (CLI Procedure)	626
Full Antivirus Scan Results and Notification Options	626
Understanding Full Antivirus Scan Result Handling	626
Protocol-Only Virus-Detected Notifications	627
Understanding Protocol-Only Virus-Detected Notifications	627
Configuring Protocol-Only Virus-Detected Notifications (CLI Procedure)	627
E-Mail Virus-Detected Notifications	627
Understanding E-Mail Virus-Detected Notifications	628
Configuring E-Mail Virus-Detected Notifications (CLI Procedure)	628
Custom Message Virus-Detected Notifications	628
Understanding Custom Message Virus-Detected Notifications	628
Configuring Custom Message Virus-Detected Notifications (CLI Procedure)	629
Full Antivirus Scanning Fallback Options	629
Understanding Antivirus Scanning Fallback Options	629
Example: Configuring Antivirus Scanning Fallback Options (CLI)	630
Full Antivirus Configuration Overview	631
Configuring Full Antivirus (J-Web Procedure)	632
Configuring Full Antivirus Custom Objects (J-Web Procedure)	632
Configuring Full Antivirus Feature Profiles (J-Web Procedure)	634
Configuring Full Antivirus UTM Policies (J-Web Procedure)	637
Attaching Full Antivirus UTM Policies to Security Policies (J-Web Procedure)	638
Example: Configuring Full Antivirus (CLI)	638
Example: Configuring Full Antivirus Custom Objects (CLI)	639
Example: Configuring Full Antivirus Feature Profiles (CLI)	640
Example: Configuring Full Antivirus UTM Policies (CLI)	643
Example: Attaching Full Antivirus UTM Policies to Security Policies (CLI)	643
Monitoring Antivirus Sessions and Scan Results	644
Monitoring Antivirus Scan Engine Status	644
Monitoring Antivirus Session Status	644
Monitoring Antivirus Scan Results	645
Chapter 32 Express Antivirus Protection	649
Express Antivirus Protection Overview	649
Express Antivirus Packet-Based Scanning Versus File-Based Scanning	649
Express Antivirus Expanded MIME Decoding Support	650
Express Antivirus Scan Result Handling	650
Express Antivirus Intelligent Prescreening	650

	Express Antivirus Limitations	650
	Express Antivirus Scanner Pattern Database	651
	Understanding Express Antivirus Scanner Pattern Updates	651
	Example: Automatically Updating Express Antivirus Patterns (J-Web)	652
	Example: Automatically Updating Express Antivirus Patterns (CLI)	652
	Manually Updating, Reloading, and Deleting Express Antivirus Patterns (CLI Procedure)	652
	Express Antivirus Configuration Overview	653
	Configuring Express Antivirus (J-Web Procedure)	653
	Configuring Express Antivirus Custom Objects (J-Web Procedure)	653
	Configuring Express Antivirus Feature Profiles (J-Web Procedure)	655
	Configuring Express Antivirus UTM Policies (J-Web Procedure)	658
	Attaching Express Antivirus UTM Policies to Security Policies (J-Web Procedure)	658
	Example: Configuring Express Antivirus (CLI)	659
	Example: Configuring Express Antivirus Custom Objects (CLI)	659
	Example: Configuring Express Antivirus Feature Profiles (CLI)	660
	Example: Configuring Express Antivirus UTM Policies (CLI)	663
	Example: Attaching Express Antivirus UTM Policies to Security Policies (CLI)	663
Chapter 33	Content Filtering	665
	Content Filtering Overview	665
	Content Filtering Protocol Support	666
	Understanding Content Filtering Protocol Support	666
	HTTP Support	666
	FTP Support	667
	E-Mail Support	667
	Specifying Content Filtering Protocols (CLI Procedure)	667
	Example: Configuring Content Filtering	668
	Content Filtering Configuration Overview	668
	Example: Configuring Content Filtering Custom Objects	669
	Example: Configuring Content Filtering Feature Profiles	671
	Example: Configuring Content Filtering UTM Policies	674
	Example: Attaching Content Filtering UTM Policies to Security Policies . .	675
	Monitoring Content Filtering Configurations	677
Chapter 34	Web Filtering	679
	Web Filtering Overview	679
	Integrated Web Filtering	680
	Understanding Integrated Web Filtering	680
	Integrated Web Filtering Process	681
	Integrated Web Filtering Cache	681
	Integrated Web Filtering Profiles	681

Profile Matching Precedence	682
Integrated Web Filtering Configuration Overview	682
Configuring Integrated Web Filtering (J-Web Procedure)	683
Configuring Integrated Web Filtering Custom Objects (J-Web Procedure)	683
Configuring Integrated Web Filtering Feature Profiles (J-Web Procedure)	684
Configuring Integrated Web Filtering UTM Policies (J-Web Procedure)	686
Attaching Integrated Web Filtering UTM Policies to Security Policies (J-Web Procedure)	687
Example: Configuring Integrated Web Filtering (CLI)	687
Example: Configuring Integrated Web Filtering Custom Objects (CLI)	687
Example: Configuring Integrated Web Filtering Feature Profiles (CLI)	689
Example: Configuring Integrated Web Filtering UTM Policies (CLI)	690
Example: Attaching Integrated Web Filtering UTM Policies to Security Policies (CLI)	691
Displaying Global SurfControl URL categories	691
Redirect Web Filtering	691
Understanding Redirect Web Filtering	691
Redirect Web Filtering Configuration Overview	692
Configuring Redirect Web Filtering (J-Web Procedure)	693
Configuring Redirect Web Filtering Custom Objects (J-Web Procedure)	693
Configuring Redirect Web Filtering Feature Profiles (J-Web Procedure)	695
Configuring Redirect Web Filtering UTM Policies (J-Web Procedure)	696
Attaching Redirect Web Filtering UTM Policies to Security Policies (J-Web Procedure)	696
Example: Configuring Redirect Web Filtering (CLI)	697
Example: Configuring Redirect Web Filtering Custom Objects (CLI)	697
Example: Configuring Redirect Web Filtering Feature Profiles (CLI)	698
Example: Configuring Redirect Web Filtering UTM Policies (CLI)	699
Example: Attaching Redirect Web Filtering UTM Policies to Security Policies (CLI)	700
Local Web Filtering	700
Understanding Local Web Filtering	700
User-Defined URL Categories	700
Local Web Filtering Process	701
Local Web Filtering Profiles	701
Profile Matching Precedence	701
Example: Configuring Local Web Filtering (CLI)	702
Example: Configuring Local Web Filtering Custom Objects (CLI)	702
Example: Configuring Local Web Filtering Feature Profiles (CLI)	703
Example: Configuring Local Web Filtering UTM Policies (CLI)	704

	Example: Attaching Local Web Filtering UTM Policies to Security Policies (CLI)	704
	Monitoring Web Filtering Configurations	704
Part 9	Attack Detection and Prevention	
Chapter 35	Attack Detection and Prevention	709
	Attack Detection and Prevention Overview	709
Chapter 36	Reconnaissance Deterrence	711
	Reconnaissance Deterrence Overview	711
	IP Address Sweeps	711
	Understanding IP Address Sweeps	711
	Example: Blocking IP Address Sweeps	712
	Port Scanning	713
	Understanding Port Scanning	713
	Example: Blocking Port Scans	714
	Network Reconnaissance Using IP Options	715
	Understanding Network Reconnaissance Using IP Options	715
	Uses for IP Packet Header Options	715
	Screen Options for Detecting IP Options Used for Reconnaissance	717
	Example: Detecting Packets That Use IP Screen Options for Reconnaissance	718
	Operating System Probes	720
	Understanding Operating System Probes	720
	TCP Headers with SYN and FIN Flags Set	720
	Understanding TCP Headers with SYN and FIN Flags Set	720
	Example: Blocking Packets with SYN and FIN Flags Set	721
	TCP Headers With FIN Flag Set and Without ACK Flag Set	722
	Understanding TCP Headers With FIN Flag Set and Without ACK Flag Set	722
	Example: Blocking Packets With FIN Flag Set and Without ACK Flag Set	723
	TCP Header with No Flags Set	724
	Understanding TCP Header with No Flags Set	724
	Example: Blocking Packets with No Flags Set	724
	Attacker Evasion Techniques	725
	Understanding Attacker Evasion Techniques	725
	Fin Scanning	726
	Understanding FIN Scans	726
	Thwarting a FIN Scan (CLI Procedure)	726
	TCP SYN Checking	726
	Understanding TCP SYN Checking	726
	Setting TCP SYN Checking (CLI Procedure)	728
	Setting Strict SYN Checking (CLI Procedure)	729
	IP Spoofing	729
	Understanding IP Spoofing	729
	Example: Blocking IP Spoofing	729

	IP Source Route Options	730
	Understanding IP Source Route Options	730
	Example: Blocking Packets with Either a Loose or a Strict Source Route Option Set	732
	Example: Detecting Packets with Either a Loose or a Strict Source Route Option Set	733
Chapter 37	Suspicious Packet Attributes	735
	Suspicious Packet Attributes Overview	735
	ICMP Fragment Protection	735
	Understanding ICMP Fragment Protection	736
	Example: Blocking Fragmented ICMP Packets (CLI)	736
	Large ICMP Packet Protection	736
	Understanding Large ICMP Packet Protection	737
	Example: Blocking Large ICMP Packets (CLI)	737
	Bad IP Option Protection	738
	Understanding Bad IP Option Protection	738
	Example: Blocking IP Packets with Incorrectly Formatted Options (CLI)	738
	Unknown Protocol Protection	739
	Understanding Unknown Protocol Protection	739
	Example: Dropping Packets Using an Unknown Protocol (CLI)	739
	IP Packet Fragment Protection	740
	Understanding IP Packet Fragment Protection	740
	Example: Dropping Fragmented IP Packets (CLI)	740
	SYN Fragment Protection	741
	Understanding SYN Fragment Protection	741
	Example: Dropping IP Packets Containing SYN Fragments (CLI)	742
Chapter 38	Denial-of-Service Attacks	743
	DoS Attack Overview	743
	Firewall DoS Attacks	743
	Firewall DoS Attacks Overview	743
	Session Table Flood Attacks	744
	Understanding Session Table Flood Attacks	744
	Understanding Source-Based Session Limits	744
	Example: Setting Source-Based Session Limits (CLI)	745

Understanding Destination-Based Session Limits	746
Example: Setting Destination-Based Session Limits (CLI)	746
SYN-ACK-ACK Proxy Flood Attacks	747
Understanding SYN-ACK-ACK Proxy Flood Attacks	747
Example: Protecting Against a SYN-ACK-ACK Proxy Flood Attack (CLI)	747
Network DoS Attacks	748
Network DoS Attacks Overview	748
SYN Flood Attacks	748
Understanding SYN Flood Attacks	748
Example: Enabling SYN Flood Protection (CLI)	753
Configuring SYN Flood Protection Options (CLI Procedure)	753
Example: Enabling SYN Flood Protection for Webservers in the DMZ (CLI)	753
SYN Cookie Protection	756
Understanding SYN Cookie Protection	756
Example: Enabling SYN Cookie Protection (CLI)	758
ICMP Flood Protection	758
Understanding ICMP Flood Attacks	758
Example: Enabling ICMP Flood Protection (CLI)	759
UDP Flood Attacks	759
Understanding UDP Flood Attacks	759
Example: Enabling UDP Flood Protection (CLI)	760
Land Attacks	760
Understanding Land Attacks	760
Example: Protecting Against a Land Attack (CLI)	761
OS-Specific DoS Attacks	761
OS-Specific DoS Attacks Overview	761
Ping of Death Attacks	762
Understanding Ping of Death Attacks	762
Example: Protecting Against a Ping of Death Attack (CLI)	763
Teardrop Attacks	763
Understanding Teardrop Attacks	763
Example: Protecting Against a Teardrop Attack (CLI)	764
WinNuke Attacks	764
Understanding WinNuke Attacks	764
Example: Protecting Against a WinNuke Attack (CLI)	765

Part 10

Chapter 39

Application Identification

Junos OS Application Identification	769
Understanding Junos OS Application Identification Services	769
Application Identification Application Package	770
Understanding Junos OS Application Identification Application Package	770
Updating Junos OS Application Identification Extracted Application Package Overview	771

Updating Junos OS Application Identification Extracted Application Package	
Manually Overview	772
Example: Updating Junos OS Application Identification Extracted Application	
Package Manually (CLI)	772
Example: Updating Junos OS Application Identification Extracted Application	
Package Automatically (CLI)	773
Example: Verifying Junos OS Application Identification Extracted Application	
Package	774
Disabling Junos OS Application Identification (CLI Procedure)	775
Junos OS Application Identification for Nested Applications	776
Understanding Junos OS Application Identification for Nested	
Applications	776
Activating Junos OS Application Identification for Nested Applications (CLI	
Procedure)	776
Junos OS Application Identification Custom Application Signature	
Definitions	777
Understanding Junos OS Application Identification Custom Application	
Definitions	777
Example: Configuring Junos OS Application Identification Custom Application	
Definitions (CLI)	777
Example: Configuring Junos OS Application Identification Custom Nested	
Application Definitions (CLI)	780
Application System Cache	782
Understanding the Application System Cache	783
Deactivating Application System Cache Information for Application	
Identification (CLI Procedure)	783
Understanding Application System Cache Information for Nested Application	
Identification	784
Deactivating Application System Cache Information for Nested Application	
Identification (CLI Procedure)	784
Verifying Application System Cache Statistics	784
Memory and Session Limits	786
Understanding Memory and Session Limit Settings for Junos OS Application	
Identification Services	786
Example: Setting Memory and Session Limits for Junos OS Application	
Identification Services (CLI)	787
Chapter 40 AppTrack Application Tracking	789
Understanding AppTrack	789
AppTrack Usage	790
Example: Configuring AppTrack (CLI)	790
Example: Verifying AppTrack Operation (CLI)	791

Part 11

Chapter 41

Chassis Cluster

Chassis Cluster	795
Chassis Cluster Overview	795
Understanding Chassis Cluster Formation	796
Chassis Cluster Redundancy Groups	797
Understanding Chassis Cluster Redundancy Groups	797
Chassis Cluster Redundancy Groups 0 Through 128	798
Understanding Chassis Cluster Redundancy Group 0: Routing Engines	798
Understanding Chassis Cluster Redundancy Groups 1 Through 128	799
Example: Configuring Chassis Cluster Redundancy Groups (CLI)	803
Verifying Chassis Cluster Redundancy Group Status	804
Chassis Cluster Redundancy Group Interface Monitoring	804
Understanding Chassis Cluster Redundancy Group Interface Monitoring	804
Example: Configuring Chassis Cluster Interface Monitoring (CLI)	805
Chassis Cluster Redundancy Group IP Address Monitoring	806
Understanding Chassis Cluster Redundancy Group IP Address Monitoring	806
Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring (CLI)	808
Understanding Chassis Cluster Monitoring of Global-Level Objects	810
Understanding SPU Monitoring	810
Understanding Flowd Monitoring	810
Understanding Cold-Sync Monitoring	811
Chassis Cluster Redundancy Group Failover	812
Understanding Chassis Cluster Redundancy Group Failover	812
Understanding Chassis Cluster Redundancy Group Manual Failover	813
Initiating a Chassis Cluster Manual Redundancy Group Failover	814
Example: Configuring Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers (CLI)	816
Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover	816
Chassis Cluster Redundant Ethernet Interfaces	817
Understanding Chassis Cluster Redundant Ethernet Interfaces	817
Example: Configuring Chassis Cluster Redundant Ethernet Interfaces (CLI)	819
Verifying Chassis Cluster Interfaces	821
Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups	822
Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups	822
Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups (CLI)	823
Example: Configuring Chassis Cluster Minimum Links (CLI)	824
Conditional Route Advertising in a Chassis Cluster	825
Understanding Conditional Route Advertising in a Chassis Cluster	825
Example: Configuring Conditional Route Advertising in a Chassis Cluster (CLI)	826

Chassis Cluster Control Plane	828
Understanding the Chassis Cluster Control Plane	828
Understanding Chassis Cluster Control Links	829
Example: Configuring Chassis Cluster Control Ports (CLI)	830
Understanding Chassis Cluster Dual Control Links	830
Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster	832
Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices	833
Understanding Chassis Cluster Control Link Heartbeats	834
Understanding Chassis Cluster Control Link Failure and Recovery	835
Example: Configuring Chassis Cluster Control Link Recovery (CLI)	837
Verifying Chassis Cluster Control Plane Statistics	837
Clearing Chassis Cluster Control Plane Statistics	838
Chassis Cluster Data Plane	838
Understanding the Chassis Cluster Data Plane	838
Understanding Session RTOs	839
Understanding Data Forwarding	839
Understanding Fabric Data Link Failure and Recovery	840
Understanding Chassis Cluster Fabric Links	840
Understanding Chassis Cluster Dual Fabric Links	841
Example: Configuring the Chassis Cluster Fabric (CLI)	842
Verifying Chassis Cluster Data Plane Interfaces	844
Verifying Chassis Cluster Data Plane Statistics	844
Clearing Chassis Cluster Data Plane Statistics	845
Consequences of Enabling Chassis Cluster	845
Understanding What Happens When Chassis Cluster Is Enabled	845
Node Interfaces on Active SRX Series Chassis Clusters	846
Node Interfaces on Active J Series Chassis Clusters	853
Management Interface on an Active Chassis Cluster	855
Fabric Interface on an Active Chassis Cluster	856
Control Interface on an Active Chassis Cluster	856
Building a Chassis Cluster	857
Connecting SRX Series Hardware to Create a Chassis Cluster	857
Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering	860
SRX Series Chassis Cluster Configuration Overview	861
Connecting J Series Hardware to Create a Chassis Cluster	863
J Series Chassis Cluster Configuration Overview	864
Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI)	866
Example: Configuring the Chassis Cluster Management Interface (CLI)	867
Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI)	869
Verifying a Chassis Cluster Configuration	869
Verifying Chassis Cluster Statistics	870
Clearing Chassis Cluster Statistics	871
Verifying Chassis Cluster Failover Status	872
Clearing Chassis Cluster Failover Status	873

Chassis Cluster Upgrades	873
Upgrading Each Device in a Chassis Cluster Separately	873
Upgrading Both Devices in a Chassis Cluster Using a Low-Impact ISSU . . .	874
Upgrading Both Devices in a Chassis Cluster Using an ISSU	874
Rolling Back Devices in a Chassis Cluster After an ISSU	875
Guarding Against Service Failure in a Chassis Cluster ISSU	875
Enabling an Automatic Chassis Cluster Node Failback After an ISSU	876
Troubleshooting Chassis Cluster ISSU Failures	876
Deciphering Mismatched Control Link Statistics During a Chassis Cluster ISSU	876
Disabling Chassis Cluster	877
Understanding Multicast Routing on a Chassis Cluster	877
Asymmetric Chassis Cluster Deployment	878
Understanding Asymmetric Routing Chassis Cluster Deployment	878
Understanding Failures in the Trust Zone Redundant Ethernet Interface	879
Understanding Failures in the Untrust Zone Interfaces	879
Example: Configuring an Asymmetric Chassis Cluster Pair (CLI)	880
Example: Configuring an Asymmetric Chassis Cluster Pair (J-Web)	881
Active/Passive Chassis Cluster Deployment (J Series Devices)	883
Understanding Active/Passive Chassis Cluster Deployment	883
Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)	884
Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) . . .	886
Active/Passive Chassis Cluster Deployment (SRX Series Devices)	888
Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster	888
Example: Configuring an SRX Series Services Gateway for the High-End as a Chassis Cluster	902
Active/Passive Chassis Cluster Deployment with an IPsec Tunnel	915
Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel	915
Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI)	917
Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web)	920
Limitations of Chassis Clustering	923

Part 12

Chapter 42

Network Address Translation

Network Address Translation	927
NAT Overview	927
Understanding NAT Rule Sets and Rules	928
NAT Rule Sets	928
NAT Rules	929

Rule Processing	929
Static NAT	930
Understanding Static NAT	930
Understanding Static NAT Rules	931
Static NAT Configuration Overview	932
Static NAT Configuration Examples	932
Example: Configuring Static NAT for Single Address Translation	932
Example: Configuring Static NAT for Subnet Translation	936
Destination NAT	941
Understanding Destination NAT	942
Understanding Destination NAT Address Pools	942
Understanding Destination NAT Rules	943
Destination NAT Configuration Overview	944
Destination NAT Configuration Examples	944
Example: Configuring Destination NAT for Single Address Translation	944
Example: Configuring Destination NAT for IP Address and Port Translation	949
Example: Configuring Destination NAT for Subnet Translation	955
Source NAT	959
Understanding Source NAT	960
Source NAT Pools	961
Understanding Source NAT Pools	961
Understanding Source NAT Pools with PAT	962
Understanding Source NAT Pools Without PAT	963
Understanding Source NAT Pools with Address Shifting	963
Understanding Persistent Addresses	964
Understanding Source NAT Rules	964
Source NAT Configuration Overview	965
Source NAT Configuration Examples	965
Example: Configuring Source NAT for Egress Interface Translation ..	966
Example: Configuring Source NAT for Single Address Translation ..	969
Example: Configuring Source NAT for Multiple Addresses with PAT ..	974
Example: Configuring Source NAT for Multiple Addresses without PAT	979
Example: Configuring Source NAT with Address Shifting	984
Example: Configuring Source NAT with Multiple Rules	989
Example: Configuring Source and Destination NAT Translations	996
Disabling Port Randomization for Source NAT (CLI Procedure)	1002
Persistent NAT	1003
Understanding Persistent NAT	1003
Understanding Session Traversal Utilities for NAT (STUN) Protocol ..	1004
Persistent NAT Configuration Overview	1005
Example: Configuring Persistent NAT with Source NAT Address Pool (CLI)	1006
Example: Configuring Persistent NAT with Interface NAT (CLI)	1007
Configuring Proxy ARP (CLI Procedure)	1008
Verifying NAT Configuration	1009

Part 13

Chapter 43

GPRS

General Packet Radio Service 1013

GPRS Overview	1013
Gp and Gn Interfaces	1014
Gi Interface	1015
Operational Modes	1015
Policy-Based GTP	1016
Understanding Policy-Based GTP	1016
Example: Enabling GTP Inspection in Policies (CLI)	1017
GTP Inspection Objects	1018
Understanding GTP Inspection Objects	1018
Example: Creating a GTP Inspection Object (CLI)	1018
GTP Message Filtering	1018
Understanding GTP Message Filtering	1018
GTP Message-Length Filtering	1019
Understanding GTP Message-Length Filtering	1019
Example: Setting GTP Message Lengths (CLI)	1019
GTP Message-Type Filtering	1019
Understanding GTP Message-Type Filtering	1019
Example: Permitting and Denying GTP Message Types (CLI)	1019
Supported GTP Message Types	1020
GTP Message-Rate Limiting	1022
Understanding GTP Message-Rate Limiting	1022
Example: Limiting the GTP Message Rate (CLI)	1022
GTP Sequence Number Validation	1023
Understanding GTP Sequence Number Validation	1023
Example: Enabling GTP Sequence Number Validation (CLI)	1023
Understanding GTP IP Fragmentation	1023
GTP Information Elements	1024
Understanding GTP Information Elements	1024
GTP APN Filtering	1024
Understanding GTP APN Filtering	1024
Example: Setting a GTP APN and a Selection Mode (CLI)	1025
GTP IMSI Prefix Filtering	1025
Understanding IMSI Prefix Filtering of GTP Packets	1025
Example: Setting a Combined IMSI Prefix and APN Filter (CLI)	1026
GTP R6 Information Elements	1026
Understanding R6 Information Elements Removal	1026
Example: Removing R6 Information Elements from GTP Messages (CLI)	1026
Supported R6 Information Elements	1027
Understanding GGSN Redirection	1030

Part 14

Index

Index	1033
-----------------	------

About This Guide

This preface provides the following guidelines for using the *Junos OS Security Configuration Guide*:

- J Series and SRX Series Documentation and Release Notes on page xxxix
- Objectives on page xl
- Audience on page xl
- Supported Routing Platforms on page xl
- Document Conventions on page xl
- Documentation Feedback on page xlii
- Requesting Technical Support on page xlii

J Series and SRX Series Documentation and Release Notes

For a list of related J Series documentation, see
<http://www.juniper.net/techpubs/software/junos-jseries/index-main.html>.

For a list of related SRX Series documentation, see
<http://www.juniper.net/techpubs/hardware/srx-series-main.html>.

If the information in the latest release notes differs from the information in the documentation, follow the *Junos OS Release Notes*.

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at
<http://www.juniper.net/techpubs/>.

Juniper Networks supports a technical book program to publish books by Juniper Networks engineers and subject matter experts with book publishers around the world. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration using the Junos operating system (Junos OS) and Juniper Networks devices. In addition, the Juniper Networks Technical Library, published in conjunction with O'Reilly Media, explores improving network security, reliability, and availability using Junos OS configuration techniques. All the books are for sale at technical bookstores and book outlets around the world. The current list can be viewed at <http://www.juniper.net/books>.

Objectives

This guide describes how to use and configure key security features on J Series Services Routers and SRX Series Services Gateways running Junos OS. It provides conceptual information, suggested workflows, and examples where applicable.

Audience

This manual is designed for anyone who installs, sets up, configures, monitors, or administers a J Series Services Router or an SRX Series Services Gateway running Junos OS. The manual is intended for the following audiences:

- Customers with technical knowledge of and experience with networks and network security, the Internet, and Internet routing protocols
- Network administrators who install, configure, and manage Internet routers

Supported Routing Platforms

This manual describes features supported on J Series Services Routers and SRX Series Services Gateways running Junos OS.

Document Conventions

Table 1 on page xl defines the notice icons used in this guide.

Table 1: Notice Icons





Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page xli defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> show chassis alarms No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces important new terms. Identifies book names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos System Basics Configuration Guide</i> RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name <i>domain-name</i>
Plain text like this	Represents names of configuration statements, commands, files, and directories; IP addresses; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none"> To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level. The console port is labeled CONSOLE.
< > (angle brackets)	Enclose optional keywords or variables.	stub <default-metric metric>;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast <i>(string1 string2 string3)</i>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Enclose a variable for which you can substitute one or more values.	community name members [community-ids]
Indentation and braces ({ })	Identify a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	

J-Web GUI Conventions

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> In the Logical Interfaces box, select All Interfaces. To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need postsales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the JTAC User Guide located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC Hours of Operation —The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>

- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://www.juniper.net/alerts/>
- Join and participate in the Juniper Networks Community Forum: <http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/> .
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, visit us at <http://www.juniper.net/support/requesting-support.html>

PART 1

Introduction to Junos OS

- Introducing Junos OS for SRX Series Services Gateways on page 3
- Understanding IPv6 Flow-Based Processing on page 45
- Introducing Junos OS for J Series Services Routers on page 69

CHAPTER 1

Introducing Junos OS for SRX Series Services Gateways

- SRX Series Services Gateways Processing Overview on page 3
- Sessions for SRX Series Services Gateways on page 7
- Debugging for SRX Series Services Gateways on page 19
- Understanding SRX Series Services Gateways Central Point Architecture on page 23
- SRX5600 and SRX5800 Services Gateways Processing Overview on page 24
- SRX3400 and SRX3600 Services Gateways Processing Overview on page 37
- SRX210 Services Gateway Processing Overview on page 40
- Limitations of Flow and Processing on page 42

SRX Series Services Gateways Processing Overview

Junos OS for SRX Series Services Gateways integrates the world-class network security and routing capabilities of Juniper Networks. Junos OS includes a wide range of packet-based filtering, class-of-service (CoS) classifiers, and traffic-shaping features as well as a rich, extensive set of flow-based security features including policies, screens, network address translation (NAT), and other flow-based services.

Traffic that enters and exits services gateway is processed according to features you configure, such as packet filters, security policies, and screens. For example, the software can determine:

- Whether the packet is allowed into the device
- Which firewall screens to apply to the packet
- The route the packet takes to reach its destination
- Which CoS to apply to the packet, if any
- Whether to apply NAT to translate the packet's IP address
- Whether the packet requires an Application Layer Gateway (ALG)

Packets that enter and exit an SRX Series device undergo both packet-based and flow-based processing:

- Flow-based packet processing treats related packets, or a stream of packets, in the same way. Packet treatment depends on characteristics that were established for the first packet of the packet stream, which is referred to as a flow.

For the distributed processing architecture of the services gateway, all flow-based processing occurs on the SPU and sampling is multi-thread aware. Packet sequencing is maintained for the sampled packets.

- Packet-based, or stateless, packet processing treats packets discretely. Each packet is assessed individually for treatment.

For the distributed processing architecture of the services gateway, some packet-based processing, such as traffic shaping, occurs on the NPU. Some packet-based processing, such as application of classifiers to a packet, occurs on the SPU.

This topic includes the following sections:

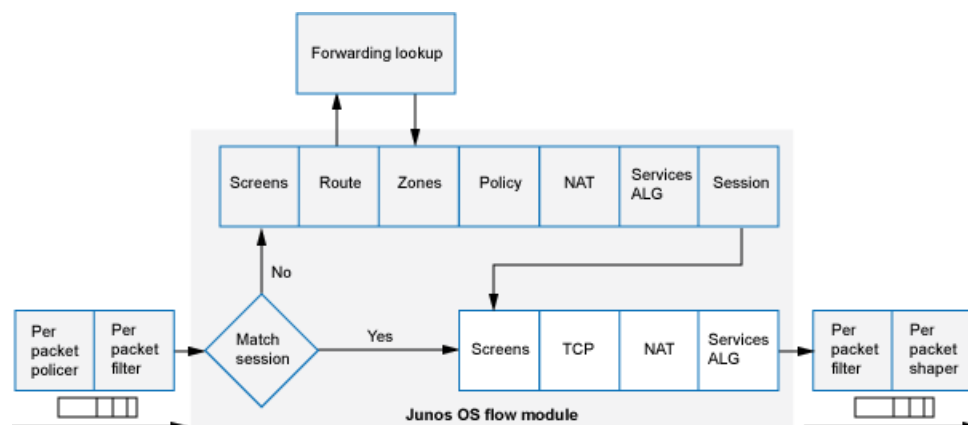
- Understanding Flow-Based Processing on page 4
- Understanding Packet-Based Processing on page 5

Understanding Flow-Based Processing

A packet undergoes flow-based processing after packet-based filters and some screens have been applied to it. All flow-based processing for a single flow occurs on a single System Processing Unit (SPU). An SPU processes the packets of a flow according to the security features and other services configured for the session.

Figure 1 on page 4 shows a conceptual view of how flow-based traffic processing occurs on services gateway.

Figure 1: Traffic Flow for Flow-Based Processing



A flow is a stream of related packets that meet the same matching criteria and share the same characteristics. Junos OS treats packets belonging to the same flow in the same manner.

Configuration settings that determine the fate of a packet—such as the security policy that applies to it, if it requires an Application Layer Gateway (ALG), if NAT is applied to translate the packet's source and/or destination IP address—are assessed for the first packet of a flow.

To determine if a flow exists for a packet, the NPU attempts to match the packet's information to that of an existing session based on the following match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Unique session token number for a given zone and virtual router

Zones and Policies

The security policy to be used for the first packet of a flow is cached in a flow table for use with the same flow and closely related flows. Security policies are associated with zones. A zone is a collection of interfaces that define a security boundary. A packet's incoming zone, as determined by the interface through which it arrived, and its outgoing zone, as determined by the forwarding lookup, together determine which policy is used for packets of the flow.

Flows and Sessions

Flow-based packet processing, which is stateful, requires the creation of sessions. A session is created for the first packet of a flow for the following purposes:

- To store most of the security measures to be applied to the packets of the flow.
- To cache information about the state of the flow.

For example, logging and counting information for a flow is cached in its session. (Some stateful firewall screens rely on threshold values that pertain to individual sessions or across all sessions.)

- To allocate required resources for the flow for features such as NAT.
- To provide a framework for features such as ALGs and firewall features.

Most packet processing occurs in the context of a flow, including:

- Management of policies, NAT, zones, and most screens.
- Management of ALGs and authentication.

Understanding Packet-Based Processing

A packet undergoes packet-based processing when it is removed from the queue from its input interface and before it is added to the queue on its output interface.

Packet-based processing applies stateless firewall filters, CoS features, and some screens to discrete packets.

- When a packet arrives at an interface, sanity checks, packet-based filters, some CoS features, and some screens are applied to it.
- Before a packet leaves the device, any packet-based filters, some CoS features, and some screens associated with the interface are applied to the packet.

Filters and CoS features are typically associated with one or more interfaces to influence which packets are allowed to transit the system and to apply special actions to packets as necessary.

The following topics describe the kinds of packet-based features that you can configure and apply to transit traffic.

Stateless Firewall Filters

Also referred to as access control lists (ACLs), stateless firewall filters control access and limit traffic rates. They statically evaluate the contents of packets transiting the device from a source to a destination, or packets originating from or destined for the Routing Engine. A stateless firewall filter evaluates every packet, including fragmented packets.

You can apply a stateless firewall filter to an input or output interface, or to both. A filter contains one or more terms, and each term consists of two components—match conditions and actions. By default, a packet that does not match a firewall filter is discarded.

You can plan and design stateless firewall filters to be used for various purposes—for example, to limit traffic to certain protocols, IP source or destination addresses, or data rates. Stateless firewall filters are executed on the SPU.

Class-of-Service Features

CoS features allow you to classify and shape traffic. CoS features are executed on the SPU.

- Behavior aggregate (BA) classifiers—These classifiers operate on packets as they enter the device. Using behavior aggregate classifiers, the device aggregates different types of traffic into a single forwarding class to receive the same forwarding treatment. BA classifiers allow you to set the forwarding class and loss priority of a packet based on the Differentiated Service (DiffServ) value.
- Traffic shaping—You can shape traffic by assigning service levels with different delay, jitter, and packet loss characteristics to particular applications served by specific traffic flows. Traffic shaping is especially useful for real-time applications, such as voice and video transmission.

Screens

Some screens, such as denial-of-service (DoS) screens, are applied to a packet outside the flow process. They are executed on the Network Processing Unit (NPU).

For details on specific stateless firewall filters and CoS features, see the *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*, the *Junos OS Class of Service Configuration Guide for Security Devices*, and the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Session Characteristics for SRX Series Services Gateways on page 7
 - SRX5600 and SRX5800 Services Gateways Processing Overview on page 24
 - SRX3400 and SRX3600 Services Gateways Processing Overview on page 37
 - SRX210 Services Gateway Processing Overview on page 40

Sessions for SRX Series Services Gateways

- Session Characteristics for SRX Series Services Gateways on page 7
- Monitoring Sessions for SRX Series Services Gateways on page 11
- Clearing Sessions for SRX Series Services Gateways on page 18

Session Characteristics for SRX Series Services Gateways

- Understanding Session Characteristics for SRX Series Services Gateways on page 7
- Example: Controlling Session Termination for SRX Series Services Gateways on page 8
- Example: Disabling TCP Packet Security Checks for SRX Series Services Gateways on page 9
- Example: Setting the Maximum Segment Size for All TCP Sessions for SRX Series Services Gateways on page 10

Understanding Session Characteristics for SRX Series Services Gateways

Sessions are created, based on routing and other classification information, to store information and allocate resources for a flow. Sessions have characteristics, some of which you can change, such as when they are terminated. For example, you might want to ensure that a session table is never entirely full to protect against an attacker's attempt to flood the table and thereby prevent legitimate users from starting sessions.

Depending on the protocol and service, a session is programmed with a timeout value. For example, the default timeout for TCP is 1800 seconds. The default timeout for UDP is 60 seconds. When a flow is terminated, it is marked as invalid, and its timeout is reduced to 10 seconds.

If no traffic uses the session before the service timeout, the session is aged out and freed to a common resource pool for reuse. You can affect the life of a session in the following ways:

- You can specify circumstances for terminating sessions by using any of the following methods:
 - Age out sessions based on how full the session table is
 - Set an explicit timeout for aging out TCP sessions

- Configure a TCP session to be invalidated when it receives a TCP RST (reset) message
- You can configure sessions to accommodate other systems as follows:
 - Disable TCP packet security checks
 - Change the maximum segment size

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- SRX Series Services Gateways Processing Overview on page 3
- Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
- Clearing Sessions for SRX Series Services Gateways on page 18
- Example: Controlling Session Termination for SRX Series Services Gateways on page 8

Example: Controlling Session Termination for SRX Series Services Gateways

This example shows how to terminate sessions for SRX Series devices based on aging out after a certain period of time, or when the number of sessions in the session table is full or reaches a specified percentage. You specify a timeout value or the number of sessions in the session table.

- Requirements on page 8
- Overview on page 8
- Configuration on page 9
- Verification on page 9

Requirements

Before you begin, understand the circumstances for terminating sessions. See “Understanding Session Characteristics for SRX Series Services Gateways” on page 7.

Overview

You can control session termination in certain situations—for example, after receiving a TCP FIN Close or receiving an RST message, when encountering ICMP errors for UDP, and when no matching traffic is received before the service timeout. When sessions are terminated, their resources are freed up for use by other sessions.

In this example, you configure the following circumstances to terminate the session:

- A timeout value of 20 seconds.
- An explicit timeout value of 280 seconds, after which the TCP session is removed from the session table.
- Any session that receives a TCP RST (reset) message is invalidated.

Configuration

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To control session termination for SRX Series devices:

1. Specify an ageout value for the session.

```
[edit security flow]  
user@host# set aging early-ageout 20
```
2. Configure an aging out value.

```
[edit security flow]  
user@host# set tcp-session tcp-initial-timeout 280
```
3. Invalidate any session that receives a TCP RST message.

```
[edit security flow]  
user@host# set tcp-session rst-invalidate-session
```
4. If you are done configuring the device, commit the configuration.

```
[edit ]  
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security flow** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Session Characteristics for SRX Series Services Gateways on page 7
- Example: Disabling TCP Packet Security Checks for SRX Series Services Gateways on page 9
- Example: Setting the Maximum Segment Size for All TCP Sessions for SRX Series Services Gateways on page 10

Example: Disabling TCP Packet Security Checks for SRX Series Services Gateways

This example shows how to disable TCP packet security checks in the device.

- Requirements on page 9
- Overview on page 10
- Configuration on page 10
- Verification on page 10

Requirements

Before you begin, understand the circumstances for disabling TCP packet security checks. See “Understanding Session Characteristics for SRX Series Services Gateways” on page 7.

Overview

Junos OS provides a mechanism for disabling security checks on TCP packets to ensure interoperability with hosts and devices with faulty TCP implementations. During no-SYN-check the Junos OS does not look for the TCP SYN packet for session creation. No-sequence check disables TCP sequence checking validation. Also, increases throughput. SYN check and sequence check are enabled by default. The set security flow command disables TCP SYN checks and TCP sequence checks on all TCP sessions thus reduces security. This may be required in scenarios with customers like big transfer files, or with applications that do not correctly work with standards.

Configuration

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#).

To disable TCP packet security checks:

1. Disable the checking of the TCP SYN bit before creating a session.

```
[edit security flow]  
user@host# set tcp-session no-syn-check
```
2. Disable the checking of sequence numbers in TCP segments during stateful inspection.

```
[edit security flow]  
user@host# set tcp-session no-sequence-check
```
3. If you are done configuring the device, commit the configuration.

```
[edit ]  
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security flow** command.

- Related Topics**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
 - [Example: Controlling Session Termination for SRX Series Services Gateways](#) on page 8
 - [Example: Setting the Maximum Segment Size for All TCP Sessions for SRX Series Services Gateways](#) on page 10

Example: Setting the Maximum Segment Size for All TCP Sessions for SRX Series Services Gateways

This example shows how to set the maximum segment size for all TCP sessions for SRX Series devices.

- [Requirements](#) on page 11
- [Overview](#) on page 11

- Configuration on page 11
- Verification on page 11

Requirements

Before you begin, understand the circumstances for setting the maximum segment size. See “Understanding Session Characteristics for SRX Series Services Gateways” on page 7.

Overview

You can terminate all TCP sessions by changing the TCP maximum segment size (TCP-MSS). To diminish the likelihood of fragmentation and to protect against packet loss, you can use the `tcp-mss` statement to specify a lower TCP MSS value. This statement applies to all IPv4 TCP SYN packets traversing all the router's ingress interfaces whose MSS value is higher than the one you specify.

If the DF bit is set, it will not fragment the packet and Junos OS will send ICMP error type 3 code 4 packet to the application server (Destination Unreachable; Fragmentation Needed and DF set). This ICMP error message contains the correct MTU (as defined in `tcp-mss`) to be used by the application server, which should receive this message and adjust the packet size accordingly. This is specifically required with VPN's since IPsec has added packet overhead, thus `tcp-mss` has to be lowered appropriately.

Configuration

Step-by-Step Procedure

To configure the maximum segment size for all TCP sessions:

1. Set the TCP maximum segment size for all TCP sessions.

[edit security flow]
user@host# **set tcp-mss all-tcp mss 1300**
2. If you are done configuring the device, commit the configuration.

[edit]
user@host# **commit**

Verification

To verify the configuration is working properly, enter the **show security flow** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Controlling Session Termination for SRX Series Services Gateways on page 8
- Example: Disabling TCP Packet Security Checks for SRX Series Services Gateways on page 9

Monitoring Sessions for SRX Series Services Gateways

- Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
- Displaying Global Session Parameters for All SRX Series Services Gateways on page 12
- Displaying a Summary of Sessions for SRX Series Services Gateways on page 13

- [Displaying Session and Flow Information About Sessions for SRX Series Services Gateways](#) on page 14
- [Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways](#) on page 14
- [Using Filters to Display Session and Flow Information for SRX Series Services Gateways](#) on page 15
- [Information Provided in Session Log Entries for SRX Series Services Gateways](#) on page 15

Understanding How to Obtain Session Information for SRX Series Services Gateways

You can obtain information about the sessions and packet flows active on your device, including detailed information about specific sessions. (The SRX Series device also displays information about failed sessions.) You can display this information to observe activity and for debugging purposes. For example, you can use the `show security flow session` command:

- To display a list of incoming and outgoing IP flows, including services
- To show the security attributes associated with a flow, for example, the policies that apply to traffic belonging to that flow
- To display the session timeout value, when the session became active, for how long it has been active, and if there is active traffic on the session

For detailed information about this command, see the *Junos OS CLI Reference*.

Session information can also be logged if a related policy configuration includes the logging option. See “Information Provided in Session Log Entries for SRX Series Services Gateways” on page 15 for details about session information provided in system logs.

Related Topics

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Session Characteristics for SRX Series Services Gateways](#) on page 7
- [Clearing Sessions for SRX Series Services Gateways](#) on page 18
- [Displaying Global Session Parameters for All SRX Series Services Gateways](#) on page 12

Displaying Global Session Parameters for All SRX Series Services Gateways

Purpose Obtain information about configured parameters that apply to all flows or sessions.

Action To view session information in the CLI, enter the following command:

```
user@host> show security flow
```

Meaning The `show security flow` configuration command displays the following information:

For detailed information about this command, see the *Junos OS CLI Reference*.

- **allow-dns-reply**—Identifies if unmatched incoming Domain Name System (DNS) reply packets are allowed.
- **route-change-timeout**—If enabled, displays the session timeout value to be used on a route change to a nonexistent route.
- **tcp-mss**—Shows the current configuration for the TCP maximum segment size value to be used for all TCP packets for network traffic.
- **tcp-session**—Displays all configured parameters that control session parameters.
- **syn-flood-protection-mode**—Displays the SYN Proxy mode.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
 - Displaying a Summary of Sessions for SRX Series Services Gateways on page 13
 - Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14
 - Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways on page 14
 - Using Filters to Display Session and Flow Information for SRX Series Services Gateways on page 15
 - Information Provided in Session Log Entries for SRX Series Services Gateways on page 15

Displaying a Summary of Sessions for SRX Series Services Gateways

Purpose Determine the kinds of sessions on your device, how many of each kind there are—for example, the number of unicast sessions and multicast sessions—the number of failed sessions, the number of sessions that are currently used and the maximum number of sessions that the device supports. This command also displays the details of the sessions that are currently used. For example, valid sessions, pending sessions, invalidated sessions and sessions in other states.

Action To view session summary information in the CLI, enter the following CLI command:

```
user@host> show security flow session summary
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
 - Displaying Global Session Parameters for All SRX Series Services Gateways on page 12
 - Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14

- [Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways](#) on page 14
- [Using Filters to Display Session and Flow Information for SRX Series Services Gateways](#) on page 15
- [Information Provided in Session Log Entries for SRX Series Services Gateways](#) on page 15

Displaying Session and Flow Information About Sessions for SRX Series Services Gateways

Purpose Display information about all sessions on your device, including the session ID, the virtual system the session belongs to, the Network Address Translation (NAT) source pool (if source NAT is used), the configured timeout value for the session and its standard timeout, and the session start time and how long the session has been active. The display also shows all standard flow information, including the direction of the flow, the source address and port, the destination address and port, the IP protocol, and the interface used for the session.

Action To view session flow information in the CLI, enter the following command:

```
user@host> show security flow session
```

- Related Topics**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
 - [Understanding How to Obtain Session Information for SRX Series Services Gateways](#) on page 12
 - [Displaying Global Session Parameters for All SRX Series Services Gateways](#) on page 12
 - [Displaying a Summary of Sessions for SRX Series Services Gateways](#) on page 13
 - [Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways](#) on page 14
 - [Using Filters to Display Session and Flow Information for SRX Series Services Gateways](#) on page 15
 - [Information Provided in Session Log Entries for SRX Series Services Gateways](#) on page 15

Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways

Purpose When you know the session identifier, you can display all session and flow information for a specific session rather than for all sessions.

Action To view information about a specific session in the CLI, enter the following command:

```
user@host> show security flow session session-identifier 40000381
```

- Related Topics**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

- Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
- Displaying Global Session Parameters for All SRX Series Services Gateways on page 12
- Displaying a Summary of Sessions for SRX Series Services Gateways on page 13
- Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14
- Using Filters to Display Session and Flow Information for SRX Series Services Gateways on page 15
- Information Provided in Session Log Entries for SRX Series Services Gateways on page 15

Using Filters to Display Session and Flow Information for SRX Series Services Gateways

Purpose You can display flow and session information about one or more sessions by specifying a filter as an argument to the **show security flow session** command. You can use the following filters: application, destination-port, destination-prefix, family, idp, interface, nat, protocol, resource-manager, session-identifier, source-port, source-prefix and tunnel. The device displays the information for each session followed by a line specifying the number of sessions reported on. Here is an example of the command using the source-prefix filter.

Action To view information about selected sessions using filters in the CLI, enter the following command:

```
user@host> show security flow session source-prefix 10/8
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
 - Displaying Global Session Parameters for All SRX Series Services Gateways on page 12
 - Displaying a Summary of Sessions for SRX Series Services Gateways on page 13
 - Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14
 - Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways on page 14
 - Information Provided in Session Log Entries for SRX Series Services Gateways on page 15

Information Provided in Session Log Entries for SRX Series Services Gateways

Session log entries are tied to policy configuration. Each main session event—create, close, and deny—will create a log entry if the controlling policy has enabled logging.

Different fields are logged for session create, session close, and session deny events as shown in Table 3 on page 16, Table 4 on page 17, and Table 5 on page 18. The same field name under each type indicates that the same information is logged, but each table is a full list of all data recorded for that type of session log.

The following table defines the fields displayed in session log entries.

Table 3: Session Create Log Fields

Field	Description
source-address	Source IP address of the packet that created the session.
source-port	Source port of the packet that created the session.
destination-address	Destination IP address of the packet that created the session.
destination-port	Destination port of the packet that created the session.
service-name	Application the packet traversed. (For example, "junos-telnet" for Telnet traffic during the session allowed by a policy that permits native Telnet.)
nat-source-address	The translated Network Address Translation (NAT) source address if NAT was applied; otherwise, the source address as above.
nat-source-port	The translated NAT source port if NAT was applied; otherwise, the source port as above.
nat-destination-address	The translated NAT destination address if NAT was applied; otherwise, the destination address as above.
nat-destination-port	The translated NAT destination port if NAT was applied; otherwise, the destination port as above.
src-nat-rule-name	The source NAT rule that was applied to the session (if any). If static NAT is also configured and applied to the session and if source address translation takes place, then this field shows the static NAT rule name.*
dst-nat-rule-name	The destination NAT rule that was applied to the session (if any). If static NAT is also configured and applied to the session and if destination address translation takes place, then this field shows the static NAT rule name.*
protocol-id	The protocol ID of the packet that created the session.
policy-name	The name of the policy that permitted the session creation.
session-id-32	The 32-bit session ID.

** Note that some sessions may have both destination and source NAT applied and the information logged.*

Table 4: Session Close Log Fields

Field	Description
reason	The reason the session was closed.
source-address	Source IP address of the packet that created the session.
source-port	Source port of the packet that created the session.
destination-address	Destination IP address of the packet that created the session.
destination-port	Destination port of the packet that created the session.
service-name	Application the packet traversed. (For example, "junos-telnet" for Telnet traffic during the session allowed by a policy that permits native Telnet.)
nat-source-address	The translated NAT source address if NAT was applied; otherwise, the source address as above.
nat-source-port	The translated NAT source port if NAT was applied; otherwise, the source port as above.
nat-destination-address	The translated NAT destination address if NAT was applied; otherwise, the destination address as above.
nat-destination-port	The translated NAT destination port if NAT was applied; otherwise, the destination port as above.
src-nat-rule-name	The source NAT rule that was applied to the session (if any). If static NAT is also configured and applied to the session and if source address translation takes place, then this field shows the static NAT rule name.*
dst-nat-rule-name	The destination NAT rule that was applied to the session (if any). If static NAT is also configured and applied to the session and if destination address translation takes place, then this field shows the static NAT rule name.*
protocol-id	The protocol ID of the packet that created the session.
policy-name	The name of the policy that permitted the session creation.
session-id-32	The 32-bit session ID.
packets-from-client	The number of packets sent by the client related to this session.
bytes-from-client	The number of data bytes sent by the client related to this session.
packets-from-server	The number of packets sent by the server related to this session.
bytes-from-server	The number of data bytes sent by the server related to this session.
elapsed-time	The total session elapsed time from permit to close, given in seconds.

Table 4: Session Close Log Fields (*continued*)

Field	Description
-------	-------------

* Note that some sessions may have both destination and source NAT applied and the information logged.

Table 5: Session Deny Log Fields

Field	Description
source-address	Source IP address of the packet that attempted to create the session.
source-port	Source port of the packet that attempted to create the session.
destination-address	Destination IP address of the packet that attempted to create the session.
destination-port	Destination port of the packet that attempted to create the session.
service-name	Application the packet attempted to traverse.
protocol-id	The protocol ID of the packet that attempted to create the session.
icmp-type	The ICMP type if the denied packet was ICMP configured; otherwise, this field will be 0.
policy-name	The name of the policy that denied the session creation.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
 - Displaying Global Session Parameters for All SRX Series Services Gateways on page 12
 - Displaying a Summary of Sessions for SRX Series Services Gateways on page 13
 - Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14
 - Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways on page 14
 - Clearing Sessions for SRX Series Services Gateways on page 18
 - Using Filters to Display Session and Flow Information for SRX Series Services Gateways on page 15

Clearing Sessions for SRX Series Services Gateways

You can use the **clear** command to terminate sessions. You can clear all sessions, including sessions of a particular application type, sessions that use a specific destination port,

sessions that use a specific interface or port, sessions that use a certain IP protocol, sessions that match a source prefix, and resource manager sessions.

- Terminating Sessions for SRX Series Services Gateways on page 19
- Terminating a Specific Session for SRX Series Services Gateways on page 19
- Using Filters to Specify the Sessions to Be Terminated for SRX Series Services Gateways on page 19

Terminating Sessions for SRX Series Services Gateways

You can use the following command to terminate all sessions except tunnel and resource manager sessions. The command output shows the number of sessions cleared. Be aware that this command terminates the management session through which the clear command is issued.

```
user@host> clear security flow session all
```

Terminating a Specific Session for SRX Series Services Gateways

You can use the following command to terminate the session whose session ID you specify.

```
user@host> clear security flow session session-identifier 40000381
```

Using Filters to Specify the Sessions to Be Terminated for SRX Series Services Gateways

You can terminate one or more sessions based on the filter parameter you specify for the **clear** command. The following example uses the protocol as a filter.

```
user@host> clear security flow session protocol 89
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12

Debugging for SRX Series Services Gateways

- Data Path Debugging for SRX Series Services Gateways on page 19
- Security Debugging for SRX Series Services Gateways on page 21
- Flow Debugging for SRX Series Services Gateways on page 22

Data Path Debugging for SRX Series Services Gateways

- Understanding Data Path Debugging for SRX Series Services Gateways on page 19
- Debugging the Data Path (CLI Procedure) on page 20

Understanding Data Path Debugging for SRX Series Services Gateways

Data path debugging provides tracing and debugging at multiple processing units along the packet-processing path. The packet filter can be executed with minimal impact to the production system.

In data path debugging, a packet goes through multiple Services Processing Units (SPUs). At the same time, several Flexible PIC Concentrator (FPC) I/O cards (IOCs) provide EZchip ingress and egress traffic management. Junos OS supports IOC for filter-based, per-packet counting and logging to record the processing path of a packet. Only the matched packets are traced by the IOC EZchip ingress, EZchip egress, load-balancing thread (LBT), and packet-ordering thread (POT).

The following events are defined in the packet-processing path:

- ezchip ingress
- ezchip egress
- spu.lbt
- spu.pot

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - SRX Series Services Gateways Processing Overview on page 3
 - Understanding Session Characteristics for SRX Series Services Gateways on page 7
 - SRX5600 and SRX5800 Services Gateways Processing Overview on page 24
 - SRX3400 and SRX3600 Services Gateways Processing Overview on page 37
 - SRX210 Services Gateway Processing Overview on page 40
 - Debugging the Data Path (CLI Procedure) on page 20

Debugging the Data Path (CLI Procedure)

To configure the device for data path debugging:

1. Specify the following request command to set the data path debugging for the multiple processing units along the packet-processing path:

user@host# set security datapath-debug
2. Specify the trace options for data path-debug using the following command:

user@host# set security datapath-debug traceoptions
3. Using the request security packet-filter command, you can set the packet filter to specify the related packets to perform data path-debug action. A maximum of four filters are supported at the same time. For example, the following command sets the first packet-filter:

user@host# set security datapath-debug packet-filter *name*
4. Using the request security action-profile command, you can set the action for the packet match for a specified filter. Only the default action profile is supported, which is the trace option for network processor ezchip ingress, ezchip egress, spu.lbt, and spu.pot:

user@host# set security datapath-debug packet-filter *name* action-profile

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Understanding Data Path Debugging for SRX Series Services Gateways on page 19
- SRX5600 and SRX5800 Services Gateways Processing Overview on page 24

Security Debugging for SRX Series Services Gateways

- Understanding Security Debugging Using Trace Options on page 21
- Setting Security Trace Options (CLI Procedure) on page 21
- Displaying Output for Security Trace Options on page 22

Understanding Security Debugging Using Trace Options

The Junos OS trace function allows applications to write security debugging information to a file. The information that appears in this file is based on criteria you set. You can use this information to analyze security application issues.

The trace function operates in a distributed manner, with each thread writing to its own trace buffer. These trace buffers are then collected at one point, sorted, and written to trace files. Trace messages are delivered using the InterProcess Communications (IPC) protocol. A trace message has a lower priority than that of control protocol packets such as BGP, OSPF, and IKE, and therefore delivery is not considered to be as reliable.

Setting Security Trace Options (CLI Procedure)

Use the following configuration statements to configure security trace options in the CLI configuration editor.

- To disable remote tracing, enter the following statement:

```
user@host# set security traceoptions no-remote-trace
```

- To write trace messages to a local file, enter the following statement. The system saves the trace file in the /var/log/ directory.

```
user@host# set security traceoptions use-local-files
```

- To specify a name for the trace file, enter the following statement. Valid values range from 1 and 1024 characters. The name cannot include spaces, /, or % characters. The default filename is security.

```
user@host# set security traceoptions file filename
```

- To specify the maximum number of trace files that can accumulate, enter the following statement. Valid values range from 2 to 1000. The default value is 3.

```
user@host# set security traceoptions file files 3
```

- To specify the match criteria that you want the system to use when logging information to the file, enter the following statement. Enter a regular expression. Wildcard (*) characters are accepted.

```
user@host# set security traceoptions file match *thread
```

- To allow any user to read the trace file, enter the **world-readable** statement. Otherwise, enter the **no-world-readable** statement.

```
user@host# set security traceoptions file world-readable
user@host# set security traceoptions file no-world-readable
```

- To specify the maximum size to which the trace file can grow, enter the following statement. Once the file reaches the specified size, it is compressed and renamed *filename0.gz*, the next file is named *filename1.gz*, and so on. Valid values range from 10240 to 1,073,741,824.

```
user@host#set security traceoptions file size 10240
```

- To turn on trace options and to perform more than one tracing operation, set the following flags.

```
user@host#set security traceoptions flag all
user@host#set security traceoptions flag compilation
user@host# set security traceoptions flag configuration
user@host#set security traceoptions flag routing-socket
```

- To specify the groups that these trace option settings do or do not apply to, enter the following statements:

```
user@host# set security traceoptions apply-groups value
user@host# set security traceoptions apply-groups-except value
```

Displaying Output for Security Trace Options

Purpose Display output for security trace options.

Action Use the **show security traceoptions** command to display the output of your trace files. For example:

```
user@host # show security traceoptions file usp_trace
user@host # show security traceoptions flag all
user@host # show security traceoptions rate-limit 888
```

The output for this example is as follows:

```
Apr 11 16:06:42 21:13:15.750395:CID-906489336:FPC-01:PIC-01:THREAD_ID-01:PFE:now
update 0x3607edf8df8in 0x3607e8d0
Apr 11 16:06:42 21:13:15.874058:CID-1529687608:FPC-01:PIC-01:THREAD_ID-01:CTRL:Enter
Function[util_ssam_handler]
Apr 11 16:06:42 21:13:15.874485:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default1: Rate
limit changed to 888
Apr 11 16:06:42 21:13:15.874538:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default1:
Destination ID set to 1
Apr 11 16:06:42 21:13:15.874651:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default2: Rate
limit changed to 888
Apr 11 16:06:42 21:13:15.874832:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default2:
Destination ID set to 1
Apr 11 16:06:42 21:13:15.874942:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default3: Rate
limit changed to 888
Apr 11 16:06:42 21:13:15.874997:CID-00:FPC-01:PIC-01:THREAD_ID-01:CTRL:default3:
Destination ID set to 1
```

Flow Debugging for SRX Series Services Gateways

- Understanding Flow Debugging Using Trace Options on page 23
- Setting Flow Debugging Trace Options (CLI Procedure) on page 23

Understanding Flow Debugging Using Trace Options

For flow trace options, you can define a packet filter using combinations of **destination-port**, **destination-prefix**, **interface**, **protocol**, **source-port**, and **source-prefix**. If the security flow trace flag for a certain module is set, the packet matching the specific packet filter triggers flow tracing and writes debugging information to the trace file.

Setting Flow Debugging Trace Options (CLI Procedure)

The following examples display the options you can set by using **security flow traceoptions**.

- To match the imap destination port for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 destination-port imap
```

- To set the 1.2.3.4 destination IPv4 prefix address for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 destination-prefix 1.2.3.4
```

- To set the fxp0 logical interface for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 interface fxp0
```

- To match the TCP IP protocol for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 protocol tcp
```

- To match the HTTP source port for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 source-port http
```

- To set the 5.6.7.8 IPv4 prefix address for the filter1 packet filter, use the following statement:

```
user@host# set security flow traceoptions packet-filter filter1 source-prefix 5.6.7.8
```

Understanding SRX Series Services Gateways Central Point Architecture

The central point (CP) in the architecture has two basic flow functionalities: load balancing and traffic identification (global session matching). The central point forwards a packet to its Services Processing Unit (SPU) upon session matching, or distributes traffic to an SPU for security processing if the packet does not match any existing session.

An SPU dedicated to central point functionality is called a large central point. However, when such a dedicated SPU is not affordable, you can configure the SPU to perform normal flow processing as well as the functions of the central point. When an SPU functions in such a dual manner, it is said to be in combination or combo mode. In combo mode, the central point and SPU share the same load-balancing thread (LBT) and packet-ordering thread (POT) infrastructure.

This topic includes the following sections:

- Load Distribution in Combo Mode on page 24
- Sharing Processing Power and Memory in Combo Mode on page 24

Load Distribution in Combo Mode

The central point maintains SPU mapping table (for load distribution) that lists live SPUs with the logic SPU IDs mapped to the physical Trivial Network Protocol (TNP) addresses mapping. In combo mode, the SPU that hosts the central point is included in the table. The load distribution algorithm is adjusted based on session capacity and processing power to avoid overloading of sessions.

Sharing Processing Power and Memory in Combo Mode

The CPU processing power in a combo-mode SPU is shared based on the platform and the number of SPUs in the system. Similarly, the CPU memory is also shared between the central point and SPU.

An SPU has multiple cores (CPUs) for networking processing. In "small" SPU combo mode, CPU functionality takes a small portion of the cores, whereas "medium" SPU combo-mode requires a larger portion of cores. The processing power for central point functionalities and flow processing is shared, based on the number of SPUs, as shown in Table 6.

Table 6: Combo Mode Processing

Number of SPUs	1	2	3	4 or More than 4
SRX3400	Small	Medium	Medium	Medium
SRX3600	Small	Medium	Medium	Medium

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
 - Understanding Session Characteristics for SRX Series Services Gateways on page 7

SRX5600 and SRX5800 Services Gateways Processing Overview

Junos OS is a distributed parallel processing high throughput and high performance system. The distributed parallel processing architecture of the services gateways includes multiple processors to manage sessions and run security and other services processing. This architecture provides greater flexibility and allows for high throughput and fast performance.



NOTE: In SRX3400, SRX3600, SRX5600, and SRX5800 devices, IKE negotiations involving NAT traversal do not work if the IKE peer is behind a NAT device that will change the source IP address of the IKE packets during the negotiation. For example, if the NAT device is configured with DIP, it changes the source IP because the IKE protocol switches the UDP port from 500 to 4500.

The SRX5600 and SRX5800 Services Gateways include I/O cards (IOC) and Services Processing Cards (SPCs) that each contain processing units that process a packet as it traverses the device. A Network Processing Unit (NPU) runs on an IOC. An IOC has one or more NPUs. One or more Services Processing Units (SPUs) run on an SPC.

These processing units have different responsibilities. All flow-based services for a packet are executed on a single SPU. Otherwise, however, the lines are not clearly divided in regard to the kinds of services that run on these processors. (For details on flow-based processing, see “Understanding Flow-Based Processing” on page 4.)

For example:

- An NPU processes packets discretely. It performs sanity checks and applies some screens that are configured for the interface, such as denial-of-service (DoS) screens, to the packet.
- An SPU manages the session for the packet flow and applies security features and other services to the packet. It also applies packet-based stateless firewall filters, classifiers, and traffic shapers to the packet.
- The system uses one processor as a central point (CP) to take care of arbitration and allocation of resources and distribute sessions in an intelligent way. The central point assigns an SPU to be used for a particular session when the first packet of its flow is processed.

These discrete, cooperating parts of the system, including the central point, each store the information identifying whether a session exists for a stream of packets and the information against which a packet is matched to determine if it belongs to an existing session.

This architecture allows the device to distribute processing of all sessions across multiple SPUs. It also allows an NPU to determine if a session exists for a packet, to check the packet, and to apply screens to it. How a packet is handled depends on whether it is the first packet of a flow.

The following sections describe the SRX5600 and SRX5800 processing architecture:

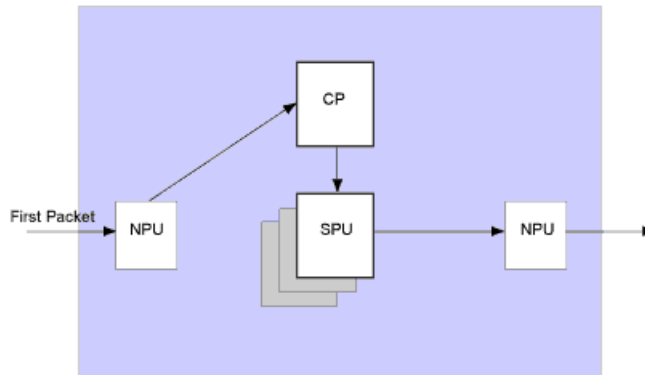
- Understanding First-Packet Processing on page 25
- Understanding Fast-Path Processing on page 27
- Understanding the Data Path for Unicast Sessions on page 28
- Understanding Packet Processing on page 34
- Understanding Services Processing Units on page 35
- Understanding Scheduler Characteristics on page 35
- Understanding Network Processor Bundling on page 35

Understanding First-Packet Processing

If the packet matches an existing flow, processing for the packet is assessed in the context of its flow state. The SPU maintains the state for each session, and the settings are then applied to the rest of the packets in the flow. If the packet does not match an existing flow, it is used to create a flow state and a session is allocated for it.

Figure 2 on page 26 illustrates the path the first packet of a flow takes as it enters the device: the NPU determines that no session exists for the packet, and the NPU sends the packet to the central point; the central point selects the SPU to set up the session for the packet and process it, and it sends the packet to that SPU. The SPU processes the packet and sends it to the NPU for transmission from the device. (This high-level description does not address application of features to a packet.)

Figure 2: First-Packet Processing



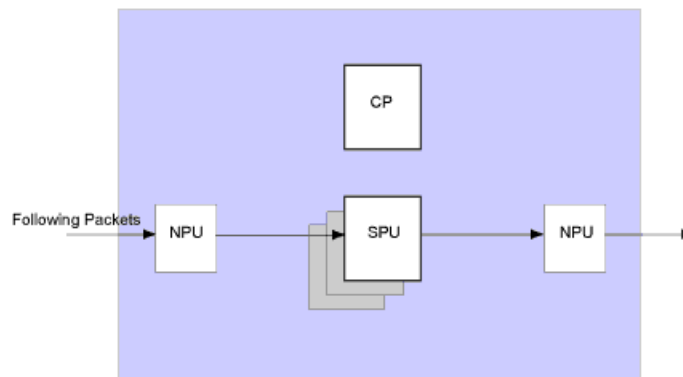
For details on session creation for the first packet in a flow, see “Understanding Session Creation: First-Packet Processing” on page 28.

After the first packet of a flow has traversed the system and a session has been established for it, it undergoes fast-path processing.

Subsequent packets of the flow also undergo fast-path processing; in this case, after each packet enters the session and the NPU finds a match for it in its session table, the NPU forwards the packet to the SPU that manages its session.

Figure 3 illustrates fast-path processing. This is the path a packet takes when a flow has already been established for its related packets. (It is also the path that the first packet of a flow takes after the session for the flow that the packet initiated has been set up.) After the packet enters the device, the NPU finds a match for the packet in its session table, and it forwards the packet to the SPU that manages the packet's session. Note that the packet bypasses interaction with the central point.

Figure 3: Fast-Path Processing



This section explains how a session is created and the process a packet undergoes as it transits the device.

Understanding Fast-Path Processing

Here is an overview of the main components involved in setting up a session for a packet and processing the packets both discretely and as part of a flow as they transit the SRX5600 and SRX5800 devices:

- Network Processing Units (NPUs)—NPUs reside on I/O cards. They handle packet sanity checking and application of some screens. NPUs maintain session tables that they use to determine if a session exists for an incoming packet or for reverse traffic.

The NPU session table contains an entry for a session if the session is established on an SPU for a packet that had previously entered the device via the interface and was processed by this NPU. The SPU installs the session in the NPU table when it creates the session.

An NPU determines if a session exists for a packet by checking the packet information against its session table. If the packet matches an existing session, the NPU sends the packet and the metadata for it to the SPU. If there is no session, the NPU sends the packet to the central point for SPU assignment.

- Services Processing Units (SPUs)—The main processors of the SRX5600 and SRX5800 devices reside on Services Processing Cards (SPCs). They establish and manage traffic flows and perform most of the packet processing on a packet as it transits the device. Each SPU maintains a hash table for fast session lookup. The SPU applies stateless firewall filters, classifiers, and traffic shapers to traffic. An SPU performs all flow-based processing for a packet and most packet-based processing. Each multicore SPU processes packets independently with minimum interaction among SPUs on the same or different SPC. All packets that belong to the same flow are processed by the same SPU.

The SPU maintains a session table with entries for all sessions that it established and whose packets it processes. When an SPU receives a packet from an NPU, it checks its session table to ensure that the packet belongs to it. It also checks its session table when it receives a packet from the central point (CP) and a message to establish a session for that packet to verify that there is not an existing session for the packet.

- Central point (CP)—The SRX Series device uses one processor as a central point to take care of arbitration and allocation of resources and distribute sessions in an intelligent way to avoid multiple SPUs from wrongly handling the same flow.

The central point's main function is to delegate session processing to one of the SPUs. If the session has not yet been established, the central point selects an SPU to establish the session for the flow, based on load- balancing criteria. If the session already exists, the central point forwards packets for that flow to the SPU hosting it. It also redirects packets to the correct SPU in the event that the NPU fails to do so.

The central point maintains a global session table with information about the owner SPU of a particular session. It functions as a central repository and resource manager for the whole system.

- Routing Engine (RE)—The Routing Engine runs the control plane.

Understanding the Data Path for Unicast Sessions

This topic describes the process of establishing a session for packets belonging to a flow that transits the device.

To illustrate session establishment and the packet “walk” including the points at which services are applied to the packets of a flow, this example uses the simple case of a unicast session.

This packet “walk” brings together the packet-based processing and flow-based processing that the Junos OS performs on the packet.

Session Lookup and Packet Match Criteria

To determine if a packet belongs to an existing flow, the device attempts to match the packet's information to that of an existing session based on the following six match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Unique token from a given zone and virtual router

Understanding Session Creation: First-Packet Processing

This topic explains how a session is set up to process the packets composing a flow. To illustrate the process, this topic uses an example with a source “a” and a destination “b”. The direction from source to destination for the packets of the flow is referred to as (a ->b). The direction from destination to source is referred to as (b->a).

Step 1. A Packet Arrives at an Interface on the Device and the NPU Processes It.

This topic describes how a packet is handled when it arrives at an SRX Series device ingress IOC.

1. The packet arrives at the device's IOC and is processed by the NPU on the card.
2. The NPU performs basic sanity checks on the packet and applies some screens configured for the interface to the packet.
3. The NPU checks its session table for an existing session for the packet. (It checks the packet's tuple against those of packets for existing sessions in its session table.)
 - a. If no existent session is found, the NPU forwards the packet to the central point.
 - b. If a session match is found, the session has already been created on an SPU that was assigned to it, so the NPU forwards the packet to the SPU for processing along with the session ID. (See "Understanding Fast-Path Processing" on page 31.)

Example: Packet (a ->b) arrives at NPU1. NPU1 performs sanity checks and applies DoS screens to the packet. NPU checks its session table for a tuple match and no existing session is found. NPU1 forwards the packet to the central point for assignment to an SPU.

Step 2. The Central Point (CP) Creates a Session with a "Pending" State.

The central point maintains a global session table that includes entries for all sessions that exist across all SPUs on the device. It participates in session creation and delegates and arbitrates session resources allocation.

This process entails the following parts:

1. The central point checks its session table and gate table to determine if a session or a gate exists for the packet it receives from the NPU. (An NPU has forwarded a packet to the central point because its table indicates there is no session for it. The central point verifies this information before allocating an SPU for the session.)
2. If there is no entry that matches the packet in either table, the central point creates a pending wing for the session and selects an SPU to be used for the session, based on its load-balancing algorithm.
3. The central point forwards the first packet of the flow to the selected SPU in a message telling it to set up a session locally to be used for the packet flow.

Example: The central point creates pending wing (a ->b) for the session. It selects SPU1 to be used for it. It sends SPU1 the (a->b) packet along with a message to create a session for it.

Step 3. The SPU Sets Up the Session.

Each SPU, too, has a session table, which contains information about its sessions. When the SPU receives a message from the central point to set up a session, it checks its session table to ensure that a session does not already exist for the packet.

1. If there is no existing session for the packet, the SPU sets up the session locally.
2. The SPU sends a message to the central point telling it to install the session.



NOTE: During first-packet processing, if NAT is enabled, the SPU allocates IP address resources for NAT. In this case, the first-packet processing for the session is suspended until the NAT allocation process is completed.

The SPU adds to the queue any additional packets for the flow that it might receive until the session has been installed.

Example: SPU1 creates the session for (a->b) and sends a message back to the central point telling it to install the pending session.

Step 4. The CP Installs the Session.

The central point receives the install message from the SPU.

1. It sets the state for the session's pending wing to active.
2. It installs the reverse wing for the session as an active wing.
3. It sends an ACK (acknowledge) message to the SPU, indicating that the session is installed.

Example: The central point receives a message from SPU1 to install the session for (a->b). It sets the session state for (a->b) wing to active. It installs the reverse wing (b->a) for the session and makes it active; this allows for delivery of packets from the reverse direction of the flow: destination (b) to be delivered to the source (a).

Step 5. The SPU Sets Up the Session on the Ingress and Egress NPUs.

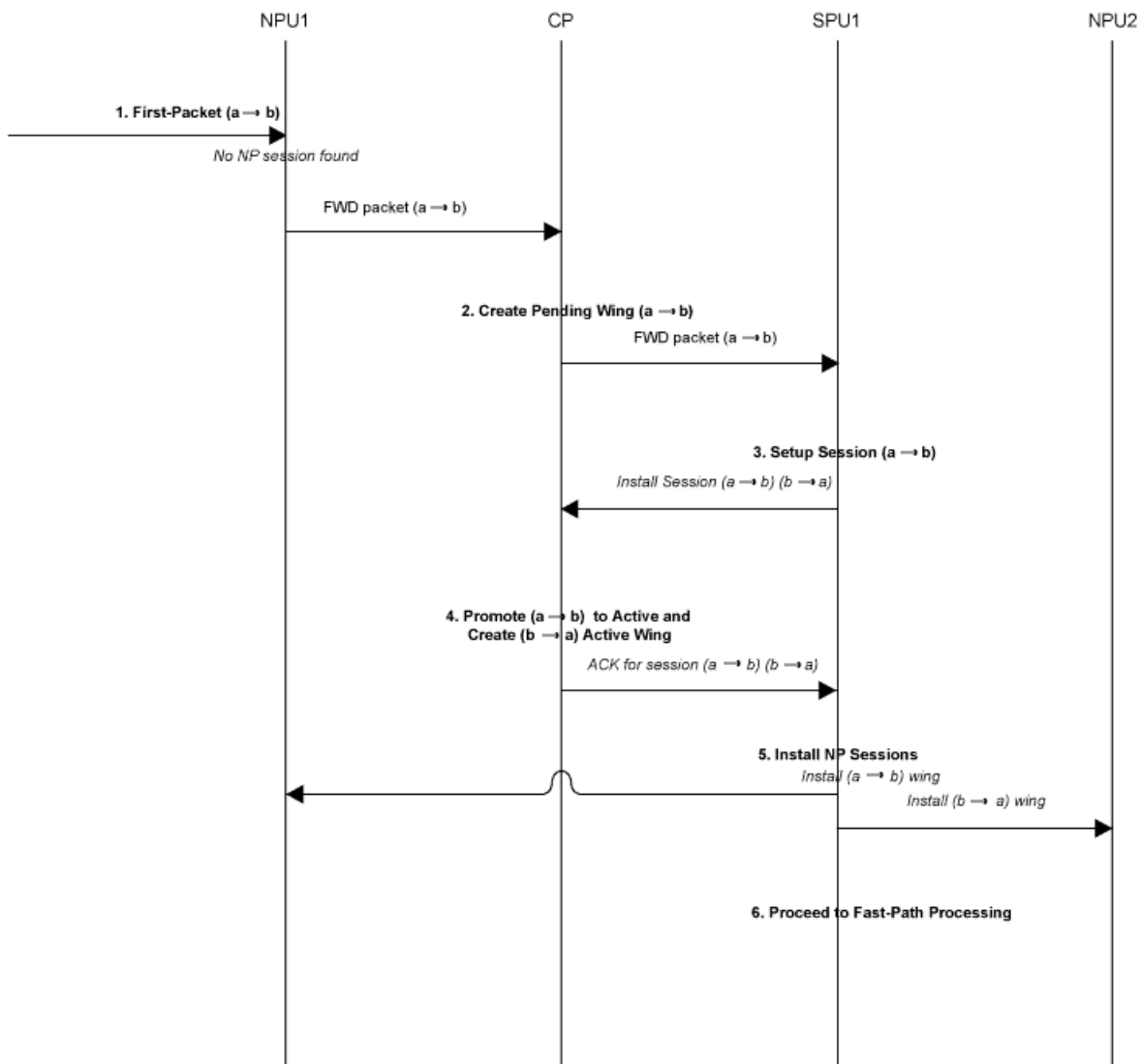
NPUs maintain information about a session for packet forwarding and delivery. Session information is set up on the egress and ingress NPUs (which sometimes are the same) so that packets can be sent directly to the SPU that manages their flows and not to the central point for redirection.

Step 6. Fast-Path Processing Takes Place.

For the remainder of the steps entailed in packet processing, proceed to Step 1 in "Understanding Fast-Path Processing" on page 31.

Figure 4 on page 31 illustrates the first part of the process the first packet of a flow undergoes after it reaches the device. At this point a session is set up to process the packet and the rest of the packets belonging to its flow. Subsequently, it and the rest of the packets of flow undergo fast-path processing.

Figure 4: Session Creation: First-Packet Processing



Understanding Fast-Path Processing

All packets undergo fast-path processing. However, if a session exists for a packet, the packet undergoes fast-path processing and bypasses the first-packet process. When there is already a session for the packet's flow, the packet does not transit the central point.

Here is how fast-path processing works: NPUs at the egress and ingress interfaces contain session tables that include the identification of the SPU that manages a packet's flow. Because the NPUs have this session information, all traffic for the flow, including reverse traffic, is sent directly to that SPU for processing.

To illustrate the fast-path process, this topic uses an example with a source "a" and a destination "b". The direction from source to destination for the packets of the flow is referred to as (a->b). The direction from destination to source is referred to as (b->a).

Step 1. A Packet Arrives at the Device and the NPU Processes It.

This topic describes how a packet is handled when it arrives at a service gateway's IOC.

1. The packet arrives at the device's IOC and is processed by the NPU on the card.
The NPU performs sanity checks and applies some screens, such as denial-of-service (DoS) screens, to the packet.
2. The NPU identifies an entry for an existing session in its session table that the packet matches.
3. The NPU forwards the packet along with metadata from its session table, including the session ID and packet tuple information, to the SPU that manages the session for the flow, applies stateless firewall filters and CoS features to its packets, and handles the packet's flow processing and application of security and other features.

Example: Packet (a ->b) arrives at NPU1. NPU1 performs sanity checks on the packet, applies DoS screens to it, and checks its session table for a tuple match. It finds a match and that a session exists for the packet on SPU1. NPU1 forwards the packet to SPU1 for processing.

Step 2. The SPU for the Session Processes the Packet.

Most of a packet's processing occurs on the SPU to which its session is assigned. The packet is processed for packet-based features such as stateless firewall filters, traffic shapers, and classifiers, if applicable. Configured flow-based security and related services such as firewall features, NAT, ALGs, and so forth, are applied to the packet. (For information on how security services are determined for a session, see "Zones and Policies" on page 5.)

1. Before it processes the packet, the SPU checks its session table to verify that the packet belongs to one of its sessions.
2. The SPU processes the packet for applicable features and services.

Example: SPU1 receives packet (a->b) from NPU1. It checks its session table to verify that the packet belongs to one of its sessions. Then it processes packet (a->b) according to input filters and CoS features that apply to its input interface. The SPU applies the security features and services that are configured for the packet's flow to it, based on its zone and policies. If any are configured, it applies output filters, traffic shapers and additional screens to the packet.

Step 3. The SPU Forwards the Packet to the NPU.

1. The SPU forwards the packet to the NPU.
2. The NPU applies any applicable screens associated with the interface to the packet.

Example: SPU1 forwards packet (a->b) to NPU2, and NPU2 applies DoS screens.

Step 4. The Interface Transmits the Packet From the Device.

Example: The interface transmits packet (a->b) from the device.

Step 5. A Reverse Traffic Packet Arrives at the Egress Interface and the NPU Processes It.

This step mirrors Step 1 exactly in reverse. See Step 1 in this topic for details.

Example: Packet (b->a) arrives at NPU2. NPU2 checks its session table for a tuple match. It finds a match and that a session exists for the packet on SPU1. NPU2 forwards the packet to SPU1 for processing.

Step 6. The SPU for the Session Processes the Reverse Traffic Packet.

This step is the same as Step 2 except that it applies to reverse traffic. See Step 2 in this topic for details.

Example: SPU1 receives packet (b->a) from NPU2. It checks its session table to verify that the packet belongs to the session identified by NPU2. Then it applies packet-based features configured for the NPU1's interface to the packet. It processes packet (b->a) according to the security features and other services that are configured for its flow, based on its zone and policies. (See "Zones and Policies" on page 5.)

Step 7. The SPU Forwards the Reverse Traffic Packet to the NPU.

This step is the same as Step 3 except that it applies to reverse traffic. See Step 3 in this topic for details.

Example: SPU1 forwards packet (b->a) to NPU1. NPU1 processes any screens configured for the interface.

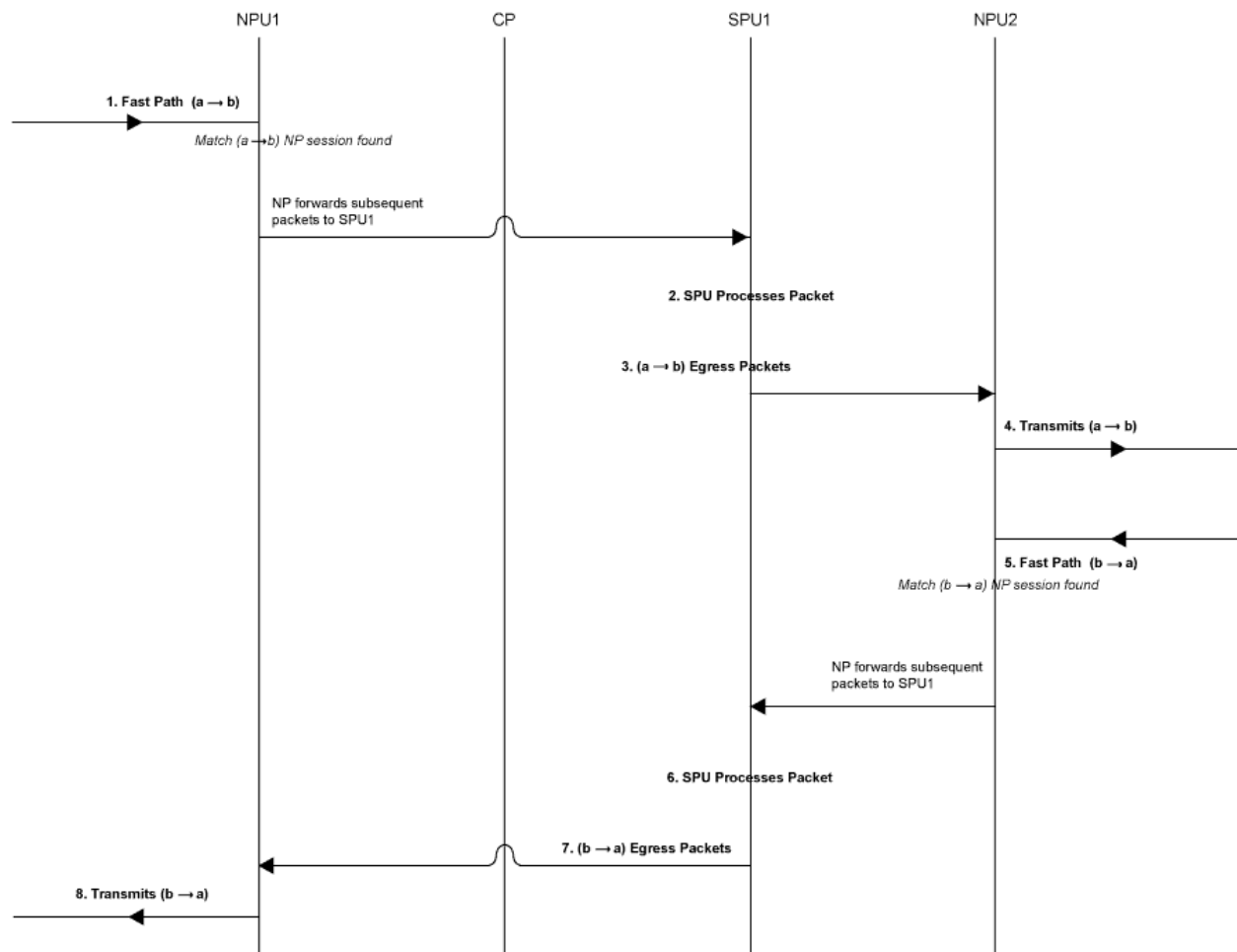
8. The Interface Transmits the Packet From the Device.

This step is the same as Step 4 except that it applies to reverse traffic. See Step 4 in this topic for details.

Example: The interface transmits packet (b->a) from the device.

Figure 5 on page 34 illustrates the process a packet undergoes when it reaches the device and a session exists for the flow that the packet belongs to.

Figure 5: "Packet Walk" for Fast-Path Processing



Understanding Packet Processing

This topic explains how a session is set up to process the packets composing a flow:

1. The primary network processor receives the IP packet and analyzes it to obtain several key tuples.
2. The interface is configured to enable distributed flow lookup and the packet is forwarded to a secondary network processor by referring a 5-tuple hash algorithm using the hash value of the primary network processor.
3. The secondary network processor receives the forwarded packet and performs all the tasks for a packet.



NOTE: When network processor bundling is enabled, screen settings are programmed into each network processor in the bundle, the settings chosen will affect the screen performance. For example, if network processor bundling is not enabled, and an ICMP screen is set to 100, ICMP packets will be blocked when the processing rate exceeds 100 pps. However, if network processor bundling is enabled and four network processors are bundled together (one primary network processor and three secondary network processors), the ICMP screen would not begin until the ICMP flood exceeds 300 pps, if the ICMP flooding packets are distributed evenly among the three secondary network processors.

Understanding Services Processing Units

For a given physical interface, the Services Processing Unit (SPU) receives ingress packets from all network processors of the network processor bundle associated to the physical interface. The SPU extracts network processor bundle information from the physical interface and uses the same 5-tuple hash algorithm to map a flow to a network processor index. To determine the network processor, the SPU does a lookup on the network processor index in the network processor bundle. The SPU sends egress packets to the physical interface's local PIC for the outward traffic.



NOTE: The network processor and the SPU use the same 5-tuple hash algorithm to get the hash values for the packets.

Understanding Scheduler Characteristics

For SRX5600 and SRX5800 devices, the IOC supports the following hierarchical scheduler characteristics:

- IFL – The configuration of the network processor bundle is stored in the physical interface data structure. The SRX5600 and SRX5800 devices have a maximum of 48 PICs. The physical interface can use a 48-bit bit-mask to indicate the PIC, or the network processor traffic from this physical interface is distributed in addition to the physical interface's primary network processor. On SRX3400, SRX3600, SRX5600, and SRX5800 devices, the iflset functionality is not supported for aggregated interfaces like *reth*.
- IFD – The logical interface associated to the physical interface of a network processor bundle is passed to all the I/O cards (IOCs) that have a PIC in the network processor bundle.

Understanding Network Processor Bundling

The network processor bundling feature is available on SRX5600 and SRX5800 Services Gateways. This feature enables distribution of data traffic from one interface to multiple network processors for packet processing. A primary network processor is assigned for an interface that receives the ingress traffic and distributes the packets to several other secondary network processors. A single network processor can act as a primary network

processor or a secondary network processor to multiple interfaces. A single network processor can join only one network processor bundle.

Network Processor Bundling Limitations

Network processor bundling functionality has the following limitations:

- Network processor bundling allows a total of 16 PICs per bundle and eight different network processor bundles system.
- You need to reboot the device to apply the configuration changes on the bundle.
- Network processor bundling is below the reth interface in the overall architecture. You can choose one or both the interfaces from the network processor bundling to form the reth interface.
- If the IOC is removed from a network processor bundle, the packets forwarded to the PIC on that IOC is lost.
- When the network processor bundle is enabled, the ICMP, UDP and TCP sync flooding thresholds no longer apply to an interface. Packets are distributed to multiple network processors for processing. These thresholds will apply to each network processor in the network processor bundle.
- Network processor bundle is not supported in the Layer 2 mode.
- Due to memory constraints on the EZchip, the number of network processor bundled ports that are supported per PIC is limited. Within the network processor bundle, each port needs to have a global port index. The global port index is calculated using the following formula:
$$\text{Global_port_index} = (\text{global_pic} * 16) + \text{port_offset}$$
- Link aggregation groups (LAGs) and redundant Ethernet interface LAGs in chassis cluster implementations can coexist with network processor bundling. However, neither LAGs nor redundant Ethernet interface LAGs can overlap with or share physical links with a network processor bundle.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Data Path Debugging for SRX Series Services Gateways on page 19
- SRX Series Services Gateways Processing Overview on page 3
- Understanding Session Characteristics for SRX Series Services Gateways on page 7
- Security Policy Schedulers Overview on page 135

SRX3400 and SRX3600 Services Gateways Processing Overview

Junos OS for the SRX3400 and SRX3600 Services Gateways integrates the world-class network security and routing capabilities of Juniper networks. Junos OS for these service gateways includes the wide range of security services including policies, screens, network address translation, class-of-service classifiers, and the rich, extensive set of flow-based services that are also supported on the other devices in the services gateways

The distributed parallel processing architecture of the SRX3400 and SRX3600 devices includes multiple processors to manage sessions and run security and other services processing. This architecture provides greater flexibility and allows for high throughput and fast performance.

This topic includes the following information:

- Components Involved in Setting up a Session on page 37
- Understanding the Data Path for Unicast Sessions on page 38
- Session Lookup and Packet Match Criteria on page 38
- Understanding Session Creation: First Packet Processing on page 38
- Understanding Fast-Path Processing on page 40

Components Involved in Setting up a Session

Here is an overview of the main components involved in setting up a session for a packet and processing the packets as they transit the SRX3400 and SRX3600 devices:

- Services Processing Units (SPUs)—The main processors of the SRX3400 and SRX3600 devices reside on Services Processing Cards (SPCs). They establish and manage traffic flows and perform most of the packet processing on a packet as it transits the device. Each SPU maintains a hash table for fast session lookup. The SPU performs all flow-based processing for a packet, including application of security services, classifiers, and traffic shapers. All packets that belong to the same flow are processed by the same SPU.

The SPU maintains a session table with entries for all sessions that it established and whose packets it processes. When an SPU receives a packet from an NPU, it checks its session table to ensure that the packet belongs to it.

For SRX3400 and SRX3600 devices, one SPU acts in concert performing its regular session management and flow processing functions and acting as a central point in which it arbitrates sessions and allocates resources. When an SPU performs in this manner it is said to be in combo mode.

- Central Point (CP)—The central point is used to allocate session management to SPUs based on load balancing criteria. It distributes sessions in an intelligent way to avoid occurrences in which multiple SPUs might wrongly handle the same flow. The central point follows load balancing criteria in allocating sessions to SPUs. If the session exists, the central point forwards packets for that flow to the SPU hosting it. It also redirects packets to the correct SPU in the event that the NPU fails to do so.

For the SRX3400 and SRX3600 devices, one SPU always runs in what is referred to as combo-mode in which it implements both the functionality of the central point and the flow and session management functionality. In combo-mode, the SPU and the central point share the same load-balancing thread (LBT) and packet-ordering thread (POT) infrastructure. For more information, see “Understanding SRX Series Services Gateways Central Point Architecture” on page 23.

- Routing Engine (RE)—The routing engine runs the control plane and manages the Control Plane Processor (CPP).

Understanding the Data Path for Unicast Sessions

Junos OS for the SRX3400 and SRX3600 Services Gateways is a distributed parallel processing high throughput and high performance system. This topic describes the process of establishing a session for packets belonging to a flow that transits the device.

To illustrate session establishment and the packet “walk” including the points at which services are applied to the packets of a flow, the following example uses the simple case of a unicast session. This packet “walk” brings together the packet-based processing and flow-based processing that the Junos OS performs on the packet.

Session Lookup and Packet Match Criteria

To determine if a packet belongs to an existing flow, the device attempts to match the packet’s information to that of an existing session based on the following six match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Unique token from a given zone and virtual router

Understanding Session Creation: First Packet Processing

This topic explains how a session is set up to process the packets composing a flow. To illustrate the process, this topic uses an example with a source “a” and a destination “b”. The direction from source to destination for the packets of the flow is referred to as (a -> b). The direction from destination to source is referred to as (b -> a).

1. A packet arrives at an interface on the device and the IOC processes it.
The IOC dequeues the packet and sends it to the NPU with which it communicates.
2. The NPU receives the packet from the IOC and processes it.
 - The NPU performs basic sanity checks on the packet and applies some screens configured for the interface to the packet.

- If a session match is found, the session has already been created on an SPU that was assigned to it, so the NPU forwards the packet to the SPU for processing along with the session ID.

Example: Packet (a ->b) arrives at NPU1 from IOC1. NPU1 performs sanity checks and applies DoS screens to the packet. NPU1 checks its session table for a tuple match and no existing session is found. NPU1 forwards the packet to the central point on SPU1 for assignment to an SPU.

3. The Central Point (CP) creates a session with a “Pending” state.

The central point maintains a global session table that includes entries for all sessions that exist across all SPUs on the device. It participates in session creation and delegates and arbitrates session resources allocation.

This process entails the following parts:

- a. The central point checks its session table and gate table to determine if a session or a gate exists for the packet it receives from the NPU. (An NPU has forwarded a packet to the central point because its table indicates there is no session for it. The central point verifies this information before allocating an SPU for the session.)
- b. If there is no entry that matches the packet in either table, the central point creates a pending wing for the session and selects an SPU to be used for the session, based on its load-balancing algorithm.
- c. The central point forwards the first packet of the flow to the selected SPU in a message telling it to set up a session locally to be used for the packet flow.

Example: The central point creates pending wing (a ->b) for the session. It selects SPU1 to be used for the session. It sends SPU1 the (a->b) packet along with a message to create a session for it. (It happens to be the case that SPU1 is the SPU that runs in combo mode. Therefore, its session-management and flow-processing services are used for the session.

4. The SPU sets up the session.

Each SPU, too, has a session table, which contains information about its sessions. When the SPU receives a message from the central point to set up a session, it checks its session table to ensure that a session does not already exist for the packet.

- a. If there is no existing session for the packet, the SPU sets up the session locally.
- b. The SPU sends a message to the central point, telling it to install the session.

During first-packet processing, if NAT is enabled, the SPU allocates IP address resources for NAT. In this case, the first-packet processing for the session is suspended until the NAT allocation process is completed.

The SPU adds to the queue any additional packets for the flow that it might receive until the session has been installed.

Example: SPU1 creates the session for (a ->b) and sends a message back to the central point (implemented on the same SPU) telling it to install the pending session.

5. The CP installs the session.

- It sets the state for the session's pending wing to active.
- It installs the reverse wing for the session as an active wing.
- It sends an ACK (acknowledge) message to the SPU, indicating that the session is installed.

Example: The central point receives a message from SPU1 to install the session for (a->b). It sets the session state for (a->b) wing to active. It installs the reverse wing (b->a) for the session and makes it active; this allows for delivery of packets from the reverse direction of the flow: destination (b) to be delivered to the source (a).

6. The SPU sets up the session on the ingress and egress NPUs.

NPUs maintain information about a session for packet forwarding and delivery. Session information is set up on the egress and ingress NPUs (which sometimes are the same) so that packets can be sent directly to the SPU that manages their flows and not to the central point for redirection.

7. Fast-path processing takes place.

For the remainder of the steps entailed in packet processing, proceed to Step 1 in "Understanding Fast-Path Processing".

Understanding Fast-Path Processing

All packets undergo fast-path processing. However, if a session exists for a packet, the packet undergoes fast-path processing and bypasses the first-packet process. When there is already a session for the packet's flow, the packet does not transit the central point.

Here is how fast-path processing works: NPUs at the egress and ingress interfaces contain session tables that include the identification of the SPU that manages a packet's flow. Because the NPUs have this session information, all traffic for the flow, including reverse traffic, is sent directly to that SPU for processing.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Data Path Debugging for SRX Series Services Gateways on page 19
- SRX Series Services Gateways Processing Overview on page 3
- Understanding Session Characteristics for SRX Series Services Gateways on page 7

SRX210 Services Gateway Processing Overview

This topic describes the process that the SRX210 Services Gateway undertakes in establishing a session for packets belonging to a flow that transits the device. The flow services of the SRX210 device are single-threaded and non-distributed. Although it differs from the other SRX Series devices in this respect, the same flow model is followed and the same command line interface (CLI) is implemented.

To illustrate session establishment and the packet “walk” including the points at which services are applied to the packets of a flow, the example described in the following sections uses the simple case of a unicast session:

- Understanding Flow Processing and Session Management on page 41
- Understanding First-Packet Processing on page 41
- Understanding Session Creation on page 41
- Understanding Fast-Path Processing on page 42

Understanding Flow Processing and Session Management

This topic explains how a session is set up to process the packets composing a flow. In the following topic, the SPU refers to the data plane thread of the SRX210 Services Gateway.

At the outset, the data plane thread fetches the packet and performs basic sanity checks on it. Then it processes the packet for stateless filters and CoS classifiers and applies some screens.

Understanding First-Packet Processing

To determine if a packet belongs to an existing flow, the device attempts to match the packet's information to that of an existing session based on the following six match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Unique token from a given zone and virtual router

The SPU checks its session table for an existing session for the packet. If no existent session is found, the SPU sets up a session for the flow. If a session match is found, the session has already been created, so the SPU performs fast-path processing on the packet.

Understanding Session Creation

In setting up the session, the SPU executes the following services for the packet:

- Screens
- Route lookup
- Policy lookup
- Service lookup
- NAT, if required

After a session is set up, it is used for all packets belonging to the flow. Packets of a flow are processed according to the parameters of its session. For the remainder of the steps entailed in packet processing, proceed to Step 1 in “Fast-Path Processing”. All packets undergo fast-path processing.

Understanding Fast-Path Processing

If a packet matches a session, Junos OS performs fast-path processing as described in the following steps. After a session has been set up for the first packet in a flow, also undergoes fast-path processing. All packets undergo fast-path processing.

1. The SPU applies flow-based security features to the packet.
 - Configured screens are applied.
 - TCP checks are performed.
 - Flow services, such as NAT, ALG, and IPsec are applied, if required.
2. The SPU prepares the packet for forwarding and transmits it.
 - Routing packet filters are applied.
 - Traffic shaping is applied.
 - Traffic prioritizing is applied.
 - Traffic scheduling is applied.
 - The packet is transmitted.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Data Path Debugging for SRX Series Services Gateways on page 19
 - SRX Series Services Gateways Processing Overview on page 3
 - Understanding Session Characteristics for SRX Series Services Gateways on page 7

Limitations of Flow and Processing

On an SRX Series or a J Series device, when defining flow and processing, be aware of the following limitations:

- On SRX3400, SRX3600, SRX5600, and SRX5800 devices, downgrading is not supported in low-impact in-service software upgrade (ISSU) chassis cluster upgrades (LICU).
- On SRX5800 devices, network processing bundling is not supported in Layer 2 transparent mode.
- On SRX210, SRX240, and J Series devices, broadcast TFTP is not supported when flow is enabled on the device.

- Maximum concurrent SSH, Telnet, and Web sessions—On SRX210, SRX240, and SRX650 devices, the maximum number of concurrent sessions is as follows:

Sessions	SRX210	SRX240	SRX650
ssh	3	5	5
telnet	3	5	5
Web	3	5	5



NOTE: These defaults are provided for performance reasons.

- On SRX210 and SRX240 devices, for optimized efficiency, we recommend that you limit use of CLI and J-Web to the following numbers of sessions:

Device	CLI	J-Web	Console
SRX210	3	3	1
SRX240	5	5	1

- On SRX100 devices, Layer 3 control protocols (OSPF, using multicast destination MAC address) on the VLAN Layer 3 interface work only with access ports.

CHAPTER 2

Understanding IPv6 Flow-Based Processing

This chapter explains how SRX Series Services Gateway and J-series devices handle flow-based processing for IP version 6 (IPv6) packets. To facilitate understanding of IPv6 flow processing for these devices, this chapter gives an overview of IPv6, including its address space and addressing. It also introduces the architecture for the SRX5600 and SRX5800 devices and uses it as a model to explain IPv6 flow processing. Flow processing is similar on other devices.

A flow is a stream of related packets that meet the same matching criteria and share the same characteristics. Junos OS treats packets belonging to the same flow in the same manner.

This chapter includes the following topics:

- Understanding IP Version 6 (IPv6) on page 46
- About the IPv6 Address Space, Addressing, and Address Types on page 46
- About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them on page 47
- About the IPv6 Address Format on page 48
- The IPv6 Packet Header and SRX Series and J-series Devices Overview on page 49
- About the IPv6 Basic Packet Header on page 50
- Understanding IPv6 Packet Header Extensions on page 52
- About IPv6 Packet Header Verification Performed by the Flow Module for SRX Series and J-series Devices on page 53
- Understanding How SRX Series and J-series Devices Handle ICMPv6 Packets on page 53
- Understanding Path MTU Messages for IPv6 Packets on page 55
- Understanding How SRX Series and J-series Devices Handle Packet Fragmentation for IPv6 Flows on page 57
- Understanding Sessions for IPv6 Flows on page 57
- Understanding SRX5600 and SRX5800 Architecture and Flow Processing on page 57
- Limitations of IPv6 on page 60

- [Enabling Flow-Based Processing for IPv6 Traffic on page 61](#)
- [Using Filters to Display IPv6 Session and Flow Information for SRX Series Services Gateways on page 63](#)

Understanding IP Version 6 (IPv6)

This topic gives an overview of IP version 6 (IPv6), including its uses and benefits.

The ongoing expansive growth of the Internet and the need to provide IP addresses to accommodate it—to support increasing numbers of new users, computer networks, Internet-enabled devices, and new and improved applications for collaboration and communication—is escalating the emergent use of a new IP protocol. IPv6, with its robust architecture, was designed to satisfy these current and anticipated near future requirements.

IP version 4 (IPv4) is widely used throughout the world today for the Internet, intranets, and private networks. IPv6 builds upon the functionality and structure of IPv4 in the following ways:

- Provides a simplified and enhanced packet header to allow for more efficient routing.
- Improves support for mobile phones and other mobile computing devices.
- Enforces increased, mandatory data security through IPsec (which was originally designed for it).
- Provides more extensive quality-of-service (QoS) support.

- Related Topics**
- [About the IPv6 Address Space, Addressing, and Address Types on page 46](#)
 - [About the IPv6 Address Format on page 48](#)
 - [About the IPv6 Basic Packet Header on page 50](#)
 - [Understanding IPv6 Packet Header Extensions on page 52](#)

About the IPv6 Address Space, Addressing, and Address Types

This topic explains IP version 6 (IPv6) addressing and identifies its three types of addresses.

Addressing is the area where most of the differences between IP version 4 (IPv4) and IPv6 exist, but the changes are largely about the ways in which addresses are implemented and used. IPv6 has a vastly larger address space than the impending exhausted IPv4 address space. IPv6 increases the size of the IP address from the 32 bits that compose an IPv4 address to 128 bits. Each extra bit given to an address doubles the size of the address space.

IPv4 has been extended using techniques such as Network Address Translation (NAT), which allows for ranges of private addresses to be represented by a single public address, and temporary address assignment. Although useful, these techniques fall short of the

requirements of novel applications and environments such as emerging wireless technologies, always-on environments, and Internet-based consumer appliances.

In addition to the increased address space, IPv6 addresses differ from IPv4 addresses in the following ways:

- Includes a scope field that identifies the type of application that the address pertains to
- Does not support broadcast addresses, but instead uses multicast addresses to broadcast a packet
- Defines a new type of address, called anycast

Related Topics

- About the IPv6 Address Format on page 48
- About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them on page 47
- About the IPv6 Basic Packet Header on page 50
- Understanding IPv6 Packet Header Extensions on page 52
- Understanding IP Version 6 (IPv6) on page 46

About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them

This topic explains the types of IP version 6 (IPv6) addresses that Junos OS for SRX Series and J-series devices support and how they are used.

IP version 6 (IPv6) includes the following types of addresses:

- Unicast

A unicast address specifies an identifier for a single interface to which packets are delivered. Under IPv6, the vast majority of Internet traffic is foreseen to be unicast, and it is for this reason that the largest assigned block of the IPv6 address space is dedicated to unicast addressing. Unicast addresses include all addresses other than loopback, multicast, link-local-unicast, and unspecified.

For SRX Series and J-series devices, the flow module supports the following kinds of IPv6 unicast packets:

- Pass-through unicast traffic, including traffic from and to virtual routers. The device transmits pass-through traffic according to its routing table.
- Host-inbound traffic from and to devices directly connected to SRX Series and J Series interfaces. For example, host-inbound traffic includes logging, routing protocol, and management types of traffic. The flow module sends these unicast packets to the Routing Engine and receives them from it. Traffic is processed by the Routing Engine instead of by the flow module, based on routing protocols defined for the Routing Engine.

The flow module supports all routing and management protocols that run on the Routing Engine. Some examples are OSPFv3, RIPng, TELNET, and SSH.

- Multicast

A multicast address specifies an identifier for a set of interfaces that typically belong to different nodes. It is identified by a value of 0xFF. IPv6 multicast addresses are distinguished from unicast addresses by the value of the high-order octet of the addresses.

The devices support only host-inbound and host-outbound multicast traffic. Host inbound traffic includes logging, routing protocols, management traffic, and so on.

- Anycast

An anycast address specifies an identifier for a set of interfaces that typically belong to different nodes. A packet with an anycast address is delivered to the nearest node, according to routing protocol rules.

There is no difference between anycast addresses and unicast addresses except for the subnet-router address. For an anycast subnet-router address, the low order bits, typically 64 or more, are zero. Anycast addresses are taken from the unicast address space.

The flow module treats anycast packets in the same way as it handles unicast packets. If an anycast packet is intended for the device, it is treated as host-inbound traffic, and it delivers it to the protocol stack which continues processing it.

Related Topics

- About the IPv6 Address Format on page 48
- Understanding IP Version 6 (IPv6) on page 46
- About the IPv6 Basic Packet Header on page 50
- Understanding IPv6 Packet Header Extensions on page 52

About the IPv6 Address Format

This topic explains the format for IP version 6 (IPv6) addresses, including how to compress them, and it gives some examples.

All IPv6 addresses are 128 bits long, written as 8 sections of 16 bits each. They are expressed in hexadecimal representation, so the sections range from 0 to FFFF. Sections are delimited by colons, and leading zeroes in each section may be omitted. If two or more consecutive sections have all zeroes, they can be collapsed to a double colon.

- IPv6 addresses have the following format in which each xxxx is a 16-bit hexadecimal value, and each x is a 4-bit hexadecimal value.

xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

- Here is an example of an IPv6 address:

3FFE:0000:0000:0001:0200:F8FF:FE75:50DF

- For an IPv6 address that contains consecutive fields of leading zeros, you can omit the zeros from each section. If you take this approach, you can write the example address in the following way:

3FFE:0:0:1:200:F8FF:FE75:50DF

- For an IPv6 address that includes contiguous sections each of which contain zeros, you can compress the 16-bit groups of zeros to double colons (::) but you can use the double-colon delimiter only once within a single IPv6 address, as shown in the following example:

3FFE::1:200:F8FF:FE75:50DF

Related Topics

- About the IPv6 Address Space, Addressing, and Address Types on page 46
- About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them on page 47
- About the IPv6 Basic Packet Header on page 50
- Understanding IPv6 Packet Header Extensions on page 52
- Understanding IP Version 6 (IPv6) on page 46

The IPv6 Packet Header and SRX Series and J-series Devices Overview

This topic identifies the IP version 6 (IPv6) packet header and its extensions and options.

Every IPv6 packet at a minimum has a basic packet header, 40 bytes (320 bits) long. They optionally may have extension headers.

For IPv6 packets, flow processing parses the extension headers and transport layer headers in the following way:

- If the software encounters a TCP, a UDP, an ESP, an AH, or an ICMPv6 header, it parses the header and assumes that the packet payload corresponds to the specified protocol type.
- If the software encounters a hop-by-hop header, a routing and destination header, or a fragment header, it continues to parse the next extension header.
- If it encounters the no-next-header extension header, the software detects that the packet is that of an unknown protocol (protocol equals 0).
- For other extension headers, the software parses the header and identifies the packet as belonging to the protocol indicated by the extension header.

Related Topics

- About the IPv6 Address Space, Addressing, and Address Types on page 46
- About the IPv6 Address Format on page 48
- About the IPv6 Basic Packet Header on page 50
- Understanding IPv6 Packet Header Extensions on page 52
- Understanding IP Version 6 (IPv6) on page 46

About the IPv6 Basic Packet Header

This topic identifies the IP version 6 (IPv6) basic packet header fields with their bit lengths and uses.

Header Name	Bit Length	Purpose
Version	4	Specifies that IP version 6 is used. The IPv6 version field contains a value of 6 indicating that IPv6 is used, as opposed to 4 for IP version 4.
Traffic Class	8	Allows source nodes or routers to identify different classes (or priorities for quality of service) for IPv6 packets. (This field replaces the IPv4 Type of Service field.)
Flow Label	20	Identifies the flow to which the packet belongs. Packets in a flow share a common purpose, or belong to a common category, as interpreted by external devices such as routers or destination hosts. NOTE: For IPv6 flow-based packets, Junos OS for SRX Series Services Gateway devices and J-series devices does not use the flow label field.
Payload Length	16	Specifies the length of the IPv6 packet payload, or contents, expressed in octets.

Header Name	Bit Length	Purpose
Next Header	8	<p>Identifies the type of Internet Protocol for the header that immediately follows the IPv6 header. The Next Header field replaces the IPv4 Protocol field. It is an optional field.</p> <p>This protocol can be one of two types:</p> <ul style="list-style-type: none"> • An IPv6 extension header. For example, if the device performs IP security on exchanged packets, the Next Header value is probably 50 (ESP extension header) or 51 (AH extension header). Extension headers are optional. • An upper-layer Protocol Data Unit (PDU). For example, the Next Header value could be 6 (for TCP), 17 (for UDP), or 58 (for ICMPv6). <p>The flow module processes these headers sequentially within the context of a packet flow.</p> <p>If it encounters one of the following extension headers, the software parses it and regards the packet as a corresponding protocol packet.</p> <ul style="list-style-type: none"> • Internet Control Message Protocol version 6 (ICMPv6) • Transport Control Protocol (TCP) <p>NOTE: The device checks the TCP header length as part of its sanity checks.</p> <ul style="list-style-type: none"> • UDP <p>NOTE: The device checks the UDP length as part of its sanity checks.</p> <ul style="list-style-type: none"> • Enhanced Security Protocol (ESP) or Authentication Header (AH)
Hop Limit	8	<p>Specifies the maximum number of hops the packet can make after transmission from the host device. When the Hop Limit value is zero, the device drops the packet and generates an error message. (This field is similar the to Time to Live IPv4 field.)</p>
Source IP Address	128	<p>Identifies the host device, or interface on a node, that generated the IPv6 packet.</p>
Destination IP Address	128	<p>Identifies the host device, or interface on a node, to which the IPv6 packet is to be sent.</p> <p>NOTE: The destination address may appear twice, the first instance after the hop limit following the source IP address and the second instance after the final extension header.</p>

- Related Topics**
- Understanding IPv6 Packet Header Extensions on page 52
 - About the IPv6 Address Space, Addressing, and Address Types on page 46
 - About the IPv6 Address Format on page 48
 - Understanding IP Version 6 (IPv6) on page 46

Understanding IPv6 Packet Header Extensions

This topic defines IP version 6 (IPv6) packet header extensions.

IPv6 extension headers contain supplementary information used by network devices (such as routers, switches, and endpoint hosts) to decide how to direct or process an IPv6 packet. The length of each extension header is an integer multiple of 8 octets. This allows subsequent extension headers to use 8-octet structures.

Any header followed by an extension header contains a Next Header value that identifies the extension header type. Extension headers always follow the basic IPv6 header in order as shown in Table 7 on page 52:



NOTE: The destination IP address can appear twice, once after the hop-by-hop header and another after the last extension header.

Table 7: IPv6 Extension Headers

Header Name	Purpose
Hop-by-Hop Options	Specifies delivery parameters at each hop on the path to the destination host. NOTE: A hop-by-hop option can appear only following the IPv6 basic header. If it is used, it should be the first extension header. It cannot appear after another extension header.
Destination Options	Specifies packet delivery parameters for either intermediate destination devices or the final destination host. When a packet uses this header, the Next Header value of the previous header must be 60.
Routing	Defines strict source routing and loose source routing for the packet. (With strict source routing, each intermediate destination device must be a single hop away. With loose source routing, intermediate destination devices can be one or more hops away.) When a packet uses this header, the Next Header value of the previous header must be 43.
Fragment	Specifies how to perform IPv6 fragmentation and reassembly services. When a packet uses this header, the Next Header value of the previous header must be 44. A source node uses the fragment extension header to tell the destination node the size of the packet that was fragmented so that the destination node can reassemble the packet.
Authentication	Provides authentication, data integrity, and anti-replay protection. When a packet uses this header, the Next Header value of the previous header must be 51.
Encapsulating Security Payload	Provides data confidentiality, data authentication, and anti-replay protection for Encapsulated Security Payload (ESP) packets. When a packet uses this header, the Next Header value of the previous header must be 50.
Destination IP Address	Identifies the host device, or interface on a node, to which the IPv6 packet is to be sent. NOTE: The destination address may appear twice, the first instance after the hop limit following the source IP address and the second instance after the final extension header.

- Related Topics**
- About the IPv6 Basic Packet Header on page 50
 - About the IPv6 Address Space, Addressing, and Address Types on page 46
 - About the IPv6 Address Format on page 48
 - Understanding IP Version 6 (IPv6) on page 46

About IPv6 Packet Header Verification Performed by the Flow Module for SRX Series and J-series Devices

This topic gives an overview of some of the IP version 6 (IPv6) packet header verification that the flow module for SRX Series and J-series devices performs.

To ensure the integrity of an IPv6 packet, the flow module performs the following sanity checks.

For all IPv6 packets, it checks the following parts of the header:

- TCP length
- UDP length
- Hop-by-hop extension to ensure that it follows the basic IPv6 header and does not come after another extension header
- That the IP data length error (IP length—total extension header length is not less than zero (<0))

In addition to these verifications, the software performs other standard checks such as verifying that the correct IP version is specified and that the length of the IP address is correct.

For details on all of the packet verification, or sanity checks, that the flow module performs, see “Understanding SRX5600 and SRX5800 Architecture and Flow Processing” on page 57

- Related Topics**
- About the IPv6 Basic Packet Header on page 50
 - Understanding IPv6 Packet Header Extensions on page 52
 - About IPv6 Address Types and How Junos OS for SRX Series Services Gateway and J-series Devices Use Them on page 47
 - About the IPv6 Address Format on page 48
 - Understanding IP Version 6 (IPv6) on page 46

Understanding How SRX Series and J-series Devices Handle ICMPv6 Packets

This topic explains Internet Control Message Protocol (ICMP), ICMP messages, and how Junos OS for SRX Series Services Gateways uses them.

ICMP provides a framework for reporting packet processing errors, for diagnostic purposes, and for implementation-specific functions. ICMP error messages make it possible for one node to inform another node that something has gone wrong during the course of data transfer. When IP version 6 (IPv6) was defined, the differences between IP version 4 (IPv4) and it were significant enough to require a new version of ICMP.

Every ICMPv6 message is preceded by an IPv6 header and zero or more IPv6 extension headers. The ICMPv6 header is identified by a Next Header value of 58 in the immediately preceding header. This is different from the value used to identify ICMP for IPv4. All ICMPv6 error messages have 32 bits of type-specific data to help the packet recipient locate the embedded invoking packet.

Most ICMPv6 packets have the same characteristics and behavior as normal IPv6 packets, and the Junos OS flow module processes them through first path and fast-path processing in the same way that it does normal IPv6 packets. Table 8 on page 54 shows the ICMPv6 embedded packet types that the flow module handles differently from normal ICMPv6 packets.

For these packets, the flow module uses a tuple that it creates from the embedded ICMPv6 packet to search for a matching session. It continues to process the packet without modifying the maximum transmission unit (MTU) until it finds a matching session, unless it receives an ICMPv6 Packet Too Big message for the interface. In this case, it modifies the MTU size for that interface. If the flow module does not find a matching session or if it cannot obtain a valid IPv6 header from the embedded payload, it drops the packet.



NOTE: A Packet Too Big message is the only kind of ICMPv6 packet that will cause the flow module to modify an interface.

Table 8: ICMPv6 Packets That Junos OS Handles Differently from Other ICMPv6 Packets

Message	Meaning
01-Destination Unreachable	<p>When a packet cannot be delivered because of a problem with the way it is being sent, it is useful to have a feedback mechanism that can tell the source about the problem, including the reason why delivery of the packet failed. For IPv6, the Destination Unreachable message serves this purpose.</p> <p>Each message includes a code that indicates the nature of the problem that caused the packet delivery to fail. It also includes all or part of the packet that could not be delivered, to help the source device resolve the problem.</p> <p>When the flow module encounters a Destination Unreachable ICMP packet whose embedded packet header data matches the 5-tuple data for a session, the software terminates the session.</p>

Table 8: ICMPv6 Packets That Junos OS Handles Differently from Other ICMPv6 Packets (*continued*)

Message	Meaning
02-Packet Too Big	<p>When the flow module receives an ICMPv6 Packet Too Big message intended for it, the flow module sends the packet to the ICMP protocol stack on the Routing Engine to engage the path maximum transmission unit (path MTU) discovery process.</p> <p>If the Packet Too Big message does not pertain to the device but rather is a transit packet, the device attempts to match the embedded 5-tuple data with a session.</p> <ul style="list-style-type: none"> • If a matching session exists, the device delivers it to the source node. • If a matching session does not exist, the device drops the packet <p>NOTE: A Packet Too Big message is the only kind of ICMPv6 packet that will cause the flow module to modify an interface.</p>
03-Time Exceeded	<p>When the flow module receives a packet that cannot be delivered because it has exceeded the hop count specified in the basic header hop-by-hop field, it sends this message to inform the packet's source node that the packet was discarded for this reason.</p>
04-Parameter Problem	<p>When the device finds a problem with a field in the IPv6 header or extension headers that makes it impossible for it to process the packet, the software discards it and sends this ICMPv6 message to the packet's source node, indicating the type and location of the problem.</p>

- Related Topics**
- Understanding Path MTU Messages for IPv6 Packets on page 55
 - Understanding How SRX Series and J-series Devices Handle Packet Fragmentation for IPv6 Flows on page 57
 - Understanding IP Version 6 (IPv6) on page 46

Understanding Path MTU Messages for IPv6 Packets

This topic describes path maximum transmission unit (MTU) and explains how the flow module for SRX Series and J-series devices processes and uses path MTU messages.

Every link has an MTU size that specifies the size of the largest packet the link can transmit. A larger MTU size means that fewer packets are required to transmit a certain amount of data. To achieve the best data transmission performance, IPv6 data packets sent from one node (the source) to another node (the destination) should be the largest possible size that can traverse the path between the nodes. (Larger and fewer packets constrain the cost of packet header processing and routing processes that can affect transmission performance.)

However, for a packet to successfully traverse the path from the source node to the destination node, the MTU size of the source node interface must be no larger than that

of the smallest MTU size of all nodes on the path between the source and destination. This value is referred to as the path maximum transmission unit (path MTU). If a packet is larger than a link's MTU size, it is likely that the link will drop it. For IPv6, an intermediate node cannot fragment a packet.

IPv6 defines a standard mechanism called path MTU discovery that a source node can use to learn the path MTU of a path that a packet is likely to traverse. If any of the packets sent on that path are too large to be forwarded by a node along the path, that node discards the packet and returns an ICMPv6 Packet Too Big message. The source node can then adjust the MTU size to be smaller than that of the node that dropped it and sent the ICMPv6 message, and then retransmit the packet. A source node might receive Packet Too Big messages repeatedly until its packet traverses all nodes along the path successfully.

After the path MTU size is determined and the appropriate MTU size is set, an outgoing packet might be routed along a different path with a node whose link MTU size is smaller than the path MTU size determined previously. In this case, the flow module engages the path MTU discovery process again.

When the flow module receives an ICMP Packet Too Big message with a destination address that belongs to it, it:

- Checks to determine if the embedded 5-tuple data of the packet is for a tunnel interface. (That is, it checks to determine if the embedded 5-tuple data matches a tunnel session.) If there is a match, the flow module updates the tunnel interface's MTU size. Then it performs post-fragment processing for the encrypted packets that follow the first packet. Afterward, the flow module delivers the packet to the ICMPv6 stack on the routing engine (RE) for it to continue processing it.
- If the packet is a transit one, the flow module searches for a session that matches the packet's embedded 5-tuple data. If it finds a matching session, it delivers the packet to it. If there is no matching session, it drops the packet.

When the flow module receives a packet, before it transmits it to the egress interface, it checks to determine if the MTU size of the egress interface is greater than the packet length.

- If the MTU size is greater than the packet length, it continues to process the packet.
- If the MTU size is less than the packet length, it drops the packet and sends an ICMPv6 Packet Too Big message to the source node.



NOTE: When chassis cluster is configured and the path MTU updates the MTU of the tunnel interface, the flow module does not sync the new MTU to peer nodes. The MTU size might be updated again by a larger packet on a peer node, which has no impact on packet transmission.

Related Topics

- Understanding How SRX Series and J-series Devices Handle ICMPv6 Packets on page 53
- About the IPv6 Basic Packet Header on page 50

- [Understanding IP Version 6 \(IPv6\) on page 46](#)

Understanding How SRX Series and J-series Devices Handle Packet Fragmentation for IPv6 Flows

This topic explains packet fragmentation for IP version 6 (IPv6).

For IPv4 Internet Control Message Protocol (IPv4 ICMP), if a node within the path between a source node and a destination node receives a packet that is larger than its MTU size, it can fragment the packet and transmit the resulting smaller packets. For IPv6, only a source node (the node that sent the packet) can fragment a packet, and this is done to accommodate a path MTU size-adjustment requirement. Nodes along the path of a packet cannot fragment the packet to transmit it.

- Related Topics**
- [Understanding How SRX Series and J-series Devices Handle ICMPv6 Packets on page 53](#)
 - [Understanding Path MTU Messages for IPv6 Packets on page 55](#)
 - [Understanding IPv6 Packet Header Extensions on page 52](#)
 - [Understanding IP Version 6 \(IPv6\) on page 46](#)

Understanding Sessions for IPv6 Flows

This topic gives an overview of flow-based sessions.

Most packet processing occurs in the context of a flow, including management of policies, zones, and most screens. A session is created for the first packet of a flow for the following purposes:

- To store most of the security measures to be applied to the packets of the flow.
- To cache information about the state of the flow. For example, logging and counting information for a flow is cached in its session. (Also, some stateful firewall screens rely on threshold values that pertain to individual sessions or across all sessions.)
- To allocate resources required for features for the flow.
- To provide a framework for features such as Application Layer Gateways (ALGs).

- Related Topics**
- [Understanding SRX5600 and SRX5800 Architecture and Flow Processing on page 57](#)
 - [Understanding IP Version 6 \(IPv6\) on page 46](#)

Understanding SRX5600 and SRX5800 Architecture and Flow Processing

This topic introduces the architecture for the SRX5600 and SRX5800 devices and uses it as a model to explain IP version 6 (IPv6) processing. Flow processing is similar on other SRX Series and J-series devices.

High-end SRX Series Services Gateway devices include I/O cards (IOCs) and Services Processing Cards (SPCs) that each contain processing units that process a packet as it traverses the device. These processing units have different responsibilities.

- A Network Processing Unit (NPU) runs on an IOC. An IOC has one or more NPUs. An NPU processes packets discretely and performs basic flow management functions.

When an IPv6 packet arrives at an IOC, the packet flow process begins. The NPU takes the following actions:

- It performs the following IPv6 sanity checks for the packet.
 - For the IPv6 basic header, it performs the following header checks:
 - Version. It verifies that the header specifies IPv6 for the version.
 - Payload length. It checks the payload length to ensure that the combined length of the IPv6 packet and the Layer 2 (L2) header is greater than the L2 frame length.
 - Hop limit. It checks to ensure that the hop limit does not specify 0 (zero)
 - Address checks. It checks to ensure that the source IP address does not specify ::0 or FF::00 and that the destination IP address does not specify ::0 or ::1.
 - It performs IPv6 extension header checks, including the following:
 - Hop-by-hop options. It verifies that this is the first extension header to follow the IPV6 basic header.
 - Routing extension. It verifies that there is only one routing extension header.
 - Destination options. It verifies that no more than two destination options extension headers are included.
 - Fragment. It verifies that there is only one fragment header.



NOTE: It treats any other extension header as a Layer 4 (L4) header.

- It performs L4 TCP, UDP, and ICMP6 protocol checks, including the following:
 - UDP. It checks to ensure that UDP packets, other than a first-fragment packet, are at least 8 bytes long.
 - TCP. It checks to ensure that ICMPv6 packets, other than a first-fragment packet, are at least 20 bytes long.
 - ICMPv6. It checks to ensure that ICMPv6 packets, other than a first-fragment packet, are at least 8 bytes long.
- If the packet specifies a TCP or a UDP protocol, it creates a tuple from the packet header data using the following information:

- Source IP address.
- Destination IP address.
- Source port.
- Destination port.
- Protocol.
- Virtual router identifier (VRID). The device looks up the VRID from a VRID table.
- For Internet Control Message Protocol version 6 (ICMPv6) packets, the tuple contains the same information as used for the TCP and the UDP search key, except for the source and destination port fields. The source and destination port fields are replaced with the following information extracted from the ICMPv6 packet:
 - For ICMP error packets: The pattern "0x00010001"
 - For ICMP information packets: The type, or code, field identifier
- For packets with an Authentication Header (AH) or an Encapsulating Security Payload (ESP) header, the search key is the same as that used for the TCP and the UDP tuple, except for the source and destination fields. In this case, the security parameter index (SPI) field value is used instead of the source and destination ports.
- If a session exists for the packet's flow, the NPU sends the packet to the SPU that manages the session.
- If a matching session does not exist,
 - The NPU sends the packet information to the central point (CP), which creates a pending session.
 - The CP selects an SPU to process the packet and create sessions for it.
 - The SPU then sends session creation messages to the CP and the ingress and egress NPUs, directing them to create a session for the packet flow.
- A central point which can run on a dedicated SPU, or share the resources of one if there is only one SPU. A CP takes care of arbitration and allocation of resources, and it distributes sessions in an intelligent way. The CP assigns an SPU to be used for a particular session when the SPU processes the first packet of its flow.
- Juniper Networks SRX5000 line devices have at least two SPUs. If an SRX5000 line device has only two SPUs, one acts in combination (*combo mode*) serving as both the CP and the SPU.
- For SRX3000 line devices, the CP and an SPU always run in combo mode.
- One or more SPUs that run on a Services Processing Card (SPC). All flow-based services for a packet are executed on a single SPU, within the context of a session that is set up for the packet flow.

The SPC for SRX5000 line devices has two SPUs. The SPC for SRX3000 line devices has one SPU.

Several SPCs can be installed in a chassis.

Primarily, an SPU performs the following tasks:

- It manages the session and applies security features and other services to the packet.
- It applies packet-based stateless firewall filters, classifiers, and traffic shapers.
- If a session does not already exist for a packet, it sends a request message to the NPU that performed the search for the packet's session, to direct it to add a session for it.

These discrete, cooperating parts of the system store the information identifying whether a session exists for a stream of packets and the information against which a packet is matched to determine if it belongs to an existing session.

- Related Topics**
- Understanding Sessions for IPv6 Flows on page 57
 - Understanding IP Version 6 (IPv6) on page 46

Limitations of IPv6

On an SRX Series or a J Series device, when defining IPv6, be aware of the following limitations:

- ALG—Application Layer Gateway (ALG) features for IPv6 sessions are not supported in Junos OS Release 10.3.
- Chassis cluster—The following features are not supported for IPv6 traffic in Junos OS Release 10.3:
 - Active-active deployments for IPv6 sessions
 - IP address monitoring for IPv6 destinations
- Class of service—Policers or simple filters for IPv6 traffic are not supported in Junos OS Release 10.3.
- Flow-based processing—If you change the forwarding option mode for IPv6, you must perform a reboot to initialize the configuration change. The following table summarizes device status upon configuration change:

Configuration Change	Commit Warning	Reboot Required	Impact on Existing Traffic Before Reboot	Impact on New Traffic Before Reboot
Drop to flow-based	Yes	Yes	Dropped	Dropped
Drop to packet-based	No	No	Packet-based	Packet-based
Flow-based to packet-based	Yes	Yes	None	Flow sessions created
Flow-based to drop	Yes	Yes	None	Flow sessions created

Packet-based to flow	Yes	Yes	Packet-based	Packet-based
Packet-based to drop	No	No	Dropped	Dropped

- IPv6 transition mechanisms—Transition mechanisms such as NAT, NAT-PT, DS-lite, or tunneling are not supported in Junos OS Release 10.3.
- J-Web—Configuration of IPv6-related settings with J-Web is not supported in Junos OS Release 10.3. You must use the CLI to configure these settings.
- Multicast—IPv6 multicast is not supported in Junos OS Release 10.3.
- NSM—Configuration of IPv6-related settings with NSM is not supported in Junos OS Release 10.3. You must use the CLI to configure these settings.
- Routing protocols—Equal cost multipath (ECMP) or Intermediate System-to-Intermediate System (IS-IS) protocols are not supported in Junos OS Release 10.3.
- Screens—The following screens are not supported for IPv6 sessions in Junos OS Release 10.3: syn-flood/syn-proxy/syn-cookie, syn-ack-ack-proxy, ip-spoofing.
- Security policy—IDP and UTM for IPv6 sessions are not supported in Junos OS Release 10.3. If your current security policy uses rules with the IP address wildcard any, and IDP and UTM features enabled, you will encounter configuration commit errors because IDP and UTM features do not yet support IPv6 addresses. To resolve the errors, modify the rule returning the error so that it uses the any-ipv4 wildcard; and create separate rules for IPv6 traffic that do not include IDP or UTM features.
- Stateless firewall filters—The following features are not supported for IPv6 traffic in Junos OS Release 10.3:
 - Matching: IPv6 prefix list
 - Actions: counter, log, reject, syslog
- System operations—DHCPv6 is not supported in Junos OS Release 10.3.
- User authentication—Firewall authentication or Web authentication over IPv6 is not supported in Junos OS Release 10.3.
- VPN—IPsec or SSL VPN for IPv6 traffic is not supported in Junos OS Release 10.3.

Enabling Flow-Based Processing for IPv6 Traffic

By default, the SRX Series or J Series device drops IPv6 traffic. To enable processing by security features such as zones, screens, and firewall policies, you must enable flow-based forwarding for IPv6 traffic.

To enable flow-based forwarding for IPv6 traffic, modify the **mode** statement at the **[edit security forwarding-options family inet6]** hierarchy level:

```
security {
  forwarding-options {
    family {
```

```

        inet6 {
            mode flow-based;
        }
    }
}

```

The following example shows the CLI commands you use to configure forwarding for IPv6 traffic.

1. Use the **set** command to change the forwarding option mode for IPv6 to flow-based.

```

[edit]
user@host# set security forwarding-options family inet6 mode flow-based

```

2. Use the **show** command to review your configuration.

```

[edit]
user@host# show security forwarding-options

family {
    inet6 {
        mode flow-based;
    }
}

```

3. Check your changes to the configuration before committing.

```

[edit]
user@host# commit check

warning: You have enabled/disabled inet6 flow.
You must reboot the system for your change to take effect.
If you have deployed a cluster, be sure to reboot all nodes.
configuration check succeeds

```

4. Commit the configuration.

```

[edit]
user@host# commit

warning: You have enabled/disabled inet6 flow.
You must reboot the system for your change to take effect.
If you have deployed a cluster, be sure to reboot all nodes.
commit complete

```

5. At an appropriate time, reboot the device.

Table 9 on page 62 summarizes device status upon forwarding option configuration change.

Table 9: Device Status Upon Configuration Change

Configuration Change	Commit Warning	Reboot Required	Impact on Existing Traffic Before Reboot	Impact on New Traffic Before Reboot
Drop to flow-based	Yes	Yes	Dropped	Dropped
Drop to packet-based	No	No	Packet-based	Packet-based

Table 9: Device Status Upon Configuration Change (*continued*)

Configuration Change	Commit Warning	Reboot Required	Impact on Existing Traffic Before Reboot	Impact on New Traffic Before Reboot
Flow-based to packet-based	Yes	Yes	None	Flow sessions created
Flow-based to drop	Yes	Yes	None	Flow sessions created
Packet-based to flow-based	Yes	Yes	Packet-based	Packet-based
Packet-based to drop	No	No	Dropped	Dropped

To process IPv6 traffic, you also need to configure IPv6 addresses for the transit interfaces that receive and forward the traffic. For information on the `inet6` protocol family and procedures for configuring IPv6 addresses for interfaces, see the *Junos OS Interfaces Configuration Guide for Security Devices*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IP Version 6 (IPv6) on page 46
 - Using Filters to Display IPv6 Session and Flow Information for SRX Series Services Gateways on page 63

Using Filters to Display IPv6 Session and Flow Information for SRX Series Services Gateways

Purpose You can display flow and session information about one or more sessions with the **show security flow session** command. IPv6 sessions are included in aggregated statistics.

You can use the following filters with the **show security flow session** command: application, destination-port, destination-prefix, family, idp, interface, nat, protocol, resource-manager, session-identifier, source-port, source-prefix and tunnel.



NOTE: Except the session-identifier filter, the output of all the other filters can be viewed in brief, summary and extensive mode. Brief mode is the default mode. The output of the session- identifier filter can be viewed only in the brief mode.

You can use the same filter options with the **clear security flow session** command to terminate sessions.

Action The following examples show how to use IPv6-related filters to display summaries and details for IPv6 sessions.

Filtered summary report based on family

```
root> show security flow session summary family ?
Possible completions:
  inet                Show IPv4 sessions
  inet6               Show IPv6/IPv6-NATPT sessions
root> show security flow session summary family inet6
```

Flow Sessions on FPC4 PIC1:

Valid sessions: 71
Pending sessions: 0
Invalidated sessions: 56
Sessions in other states: 0
Total sessions: 127

Flow Sessions on FPC5 PIC0:

Valid sessions: 91
Pending sessions: 0
Invalidated sessions: 53
Sessions in other states: 0
Total sessions: 144

Flow Sessions on FPC5 PIC1:

Valid sessions: 91
Pending sessions: 0
Invalidated sessions: 54
Sessions in other states: 0
Total sessions: 145

**Filtered detailed report
based on family**

```

root> show security flow session family ?
Possible completions:
  inet          Show IPv4 sessions
  inet6         Show IPv6/IPv6-NATPT sessions
root> show security flow session family inet6
Flow Sessions on FPC4 PIC1:

Session ID: 170001887, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 4000::100/9 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104

  Out: 4000::200/27490 --> 4000::100/9;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1

Flow Sessions on FPC5 PIC0:

Session ID: 200001865, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 4000::100/10 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104

  Out: 4000::200/27490 --> 4000::100/10;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1

Flow Sessions on FPC5 PIC1:

Session ID: 210001865, Policy name: self-traffic-policy/1, Timeout: 4, Valid
  In: 4000::100/11 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104

  Out: 4000::200/27490 --> 4000::100/11;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1

```

**Filtered brief report
based on family**

```

root> show security flow session family inet brief
Flow Sessions on FPC4 PIC1:

Session ID: 170067516, Policy name: self-traffic-policy/1, Timeout: 4, Valid
  In: 40.0.0.100/23 --> 40.0.0.1/26637;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/26637 --> 40.0.0.100/23;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1

Flow Sessions on FPC5 PIC0:

Session ID: 200066737, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 40.0.0.100/21 --> 40.0.0.1/26637;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/26637 --> 40.0.0.100/21;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1

Flow Sessions on FPC5 PIC1:

Session ID: 210066726, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 40.0.0.100/22 --> 40.0.0.1/26637;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/26637 --> 40.0.0.100/22;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1

```

**Filtered detailed report
based on an IPv6
source-prefix**

```

root> show security flow session source-prefix 4000::100
Flow Sessions on FPC4 PIC1:

Session ID: 170001907, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 4000::100/69 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104

  Out: 4000::200/27490 --> 4000::100/69;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1

Flow Sessions on FPC5 PIC0:

```

```

Session ID: 200001885, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 4000::100/70 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104

  Out: 4000::200/27490 --> 4000::100/70;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1

```

Flow Sessions on FPC5 PIC1:

```

Session ID: 210001885, Policy name: self-traffic-policy/1, Timeout: 4, Valid
  In: 4000::100/71 --> 4000::200/27490;icmp6, If: ge-0/0/2.0, Pkts: 1, Bytes: 104

  Out: 4000::200/27490 --> 4000::100/71;icmp6, If: .local..0, Pkts: 1, Bytes: 104
Total sessions: 1

```

**Multiple-filtered
detailed report based
on family, protocol and
source-prefix**

```

root> show security flow session family inet protocol icmp source-prefix 40/8
Flow Sessions on FPC4 PIC1:

```

```

Session ID: 170029413, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 40.0.0.100/50 --> 40.0.0.1/1369;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/1369 --> 40.0.0.100/50;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1

```

Flow Sessions on FPC5 PIC0:

```

Session ID: 200029073, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 40.0.0.100/51 --> 40.0.0.1/1369;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/1369 --> 40.0.0.100/51;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1

```

Flow Sessions on FPC5 PIC1:

```

Session ID: 210029083, Policy name: self-traffic-policy/1, Timeout: 2, Valid
  In: 40.0.0.100/52 --> 40.0.0.1/1369;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
  Out: 40.0.0.1/1369 --> 40.0.0.100/52;icmp, If: .local..0, Pkts: 1, Bytes: 84
Total sessions: 1

```

**Clearing all sessions,
including IPv6 sessions**

```

root> clear security flow session all
This command may terminate the current session too.
Continue? [yes,no] (no) yes
0 active sessions cleared
1 active sessions cleared
1 active sessions cleared
1 active sessions cleared

```

**Clearing only IPv6
sessions**

```

root> clear security flow session family ?
Possible completions:
  inet          Clear IPv4 sessions
  inet6         Clear IPv6/IPv6-NATPT sessions
root> clear security flow session family inet6
0 active sessions cleared
1 active sessions cleared
1 active sessions cleared
1 active sessions cleared

```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Enabling Flow-Based Processing for IPv6 Traffic on page 61
 - Understanding How to Obtain Session Information for SRX Series Services Gateways on page 12
 - Displaying a Summary of Sessions for SRX Series Services Gateways on page 13
 - Displaying Session and Flow Information About Sessions for SRX Series Services Gateways on page 14
 - Displaying Session and Flow Information About a Specific Session for SRX Series Services Gateways on page 14
 - Information Provided in Session Log Entries for SRX Series Services Gateways on page 15
 - Clearing Sessions for SRX Series Services Gateways on page 18

CHAPTER 3

Introducing Junos OS for J Series Services Routers

- Understanding Stateful and Stateless Data Processing for J Series Services Routers on page 69
- Session Characteristics for J Series Services Routers on page 73
- Understanding the Data Path for J Series Services Routers on page 79

Understanding Stateful and Stateless Data Processing for J Series Services Routers

Junos OS for J Series Services Routers integrates the world-class network security and routing capabilities of Juniper Networks Operating System.

Traffic that enters and exits a services router running Junos OS is processed according to features you configure, such as security policies, packet filters, and screens. For example, the software can determine:

- Whether the packet is allowed into the router
- Which class of service (CoS) to apply to the packet, if any
- Which firewall screens to apply to the packet
- Whether to send the packet through an IPsec tunnel
- Whether the packet requires an Application Layer Gateway (ALG)
- Whether to apply Network Address Translation (NAT) to translate the packet's address
- Which route the packet uses to reach its destination

Packets that enter and exit a services router running Junos OS undergo both packet-based and flow-based processing. A device always processes packets discretely. Packet treatment depends on characteristics that were established for the first packet of the packet stream.

Branch devices implement both packet-based and flow-based modes, concurrently. Flow-based and packet-based processing are described in the following sections:

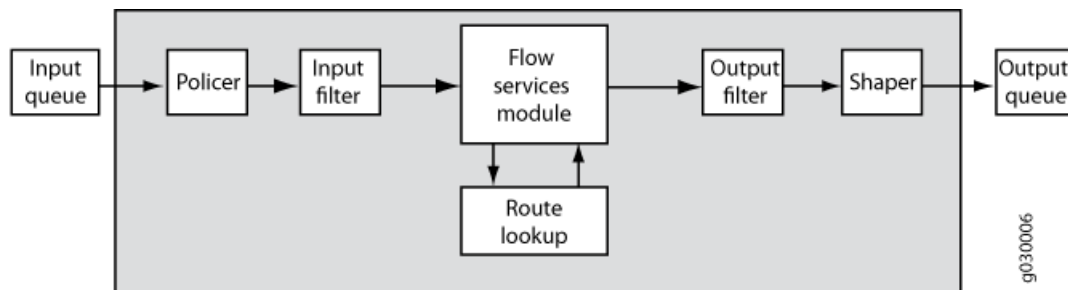
- Understanding Flow-Based Processing on page 70
- Understanding Packet-Based Processing on page 71

Understanding Flow-Based Processing

A packet undergoes flow-based processing after any packet-based filters and policers have been applied to it.

Figure 6 on page 70 shows an architectural overview of traffic flow in a Juniper Networks device running Junos OS. See Figure 8 on page 80 to follow the path of the traffic as it traverses through the flow services module.

Figure 6: Traffic Flow for Flow-Based Processing



A flow is defined as a set of packets coming from the same source/destination addresses, source/destination ports (when applicable), protocol, and ingress/egress zones. Flows are time bound so it is possible to have packets that, while fitting the previous definition, belong to different flows. For example, when an existing session is initiated and terminated, after which a new session is established using the exact same parameters as the previous session, the packets would belong to different flows.

Configuration settings that determine the fate of a packet—such as the security policy that applies to it, whether the packet is sent through an IPsec tunnel, if it requires an Application Layer Gateway (ALG), if Network Address Translation (NAT) is applied to translate the packet's address—are assessed for the first packet of a flow. The settings are then applied to the rest of the packets in the flow.

To determine if a packet belongs to an existing flow, the router attempts to match the packet's information to that of an existing flow based on the following six match criteria:

- Source address
- Destination address
- Source port
- Destination port
- Protocol
- Session token—An internal parameter not extracted from the packet's header

If the packet matches an existing flow, processing for the packet is determined by the flow state (maintained by the flow's session). If the packet does not match the session for an existing flow, the packet's information is used to create a new flow state and a session is allocated for it (a session is allocated only if this is permitted by the security

policy). Sessions used for the first packet of a flow is cached in a flow table for use with the same flow and closely related flows.



NOTE: A new session is allocated for the new flow state only if this is permitted by the security policy. For TCP, only SYN packets will trigger creating a new session (unless SYN checking is not enabled).

Zones and Policies

Security policies are associated with zones. A zone is a collection of interfaces that define a security boundary. A packet's incoming zone, as determined by the interface through which it arrived, and its outgoing zone, as determined by the forwarding lookup, together determine which policy is used for packets of the flow.

Flows and Sessions

Flow-based packet processing, which is stateful, requires the creation of sessions. A session is created, based on the characteristics assessed for the first packet of a flow, for the following purposes:

- To store the security measures to be applied to the packets of the flow
- To cache information about the state of the flow

For example, logging and counting information for a flow is cached in its session. (Some stateful firewall screens rely on threshold values that pertain to individual sessions or across all sessions.)

- To allocate required resources for the flow for features such as Network Address Translation (NAT) and IPsec tunnels
- To provide a framework for features such as Application Layer Gateways (ALGs) and firewall features

Most packet processing occurs in the context of a flow. The flow engine and session bring together the following features and events that affect a packet as it undergoes flow-based processing:

- Flow-based forwarding
- Session management, including session aging and changes in routes, policy, and interfaces
- Management of virtual private networks (VPNs), ALGs, and authentication
- Management of policies, NAT, zones, and screens

Policies can be configured to log session permit, close, and deny events.

Understanding Packet-Based Processing

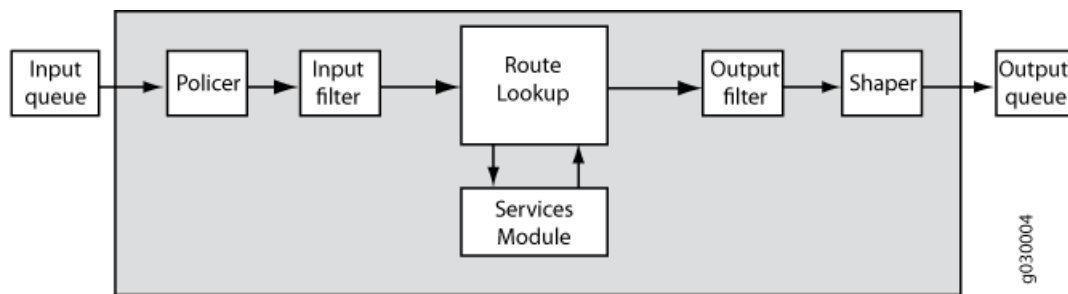
A packet undergoes packet-based processing when it is dequeued from its input (ingress) interface and before it is enqueued on its output (egress) interface.

Packet-based processing applies stateless firewall filters and class-of-service (CoS) features to discrete packets. You can apply a firewall filter to an ingress or egress interface, or to both.

- When a packet arrives at an interface on the device, any packet-based filters and policers associated with the interface are applied to the packet before any security policies are evaluated.
- Before a packet leaves the device, any packet-based filters and traffic shapers associated with the output interface are applied to the packet after any security policies have been evaluated.

Figure 7 on page 72 shows an architectural overview of traffic flow in a Juniper Networks device running Junos OS.

Figure 7: Traffic Flow for Packet-Based Processing



Filters and CoS features are typically associated with one or more interfaces to influence which packets are allowed to transit the system and to apply special actions to packets as necessary.



NOTE: Packet-based processing occurs only if you configure filters, CoS, IPv6, and MPLS features for an interface that handles the packet.

The following sections describe the kinds of packet-based features that you can configure and apply to transit traffic. For details on specific stateless firewall filters and CoS features, see the *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*, the *Junos OS Class of Service Configuration Guide for Security Devices*, and the *Junos OS CLI Reference*.

Stateless Firewall Filters

Also referred to as access control lists (ACLs), stateless firewall filters control access and limit traffic rates. They statically evaluate the contents of packets transiting the device from a source to a destination, or packets originating from or destined for the Routing Engine. A stateless firewall filter evaluates every packet, including fragmented packets.

You can apply a stateless firewall filter to an input or output interface, or to both. A filter contains one or more terms, and each terms consists of two components—match conditions and actions. By default, a packet that does not match a firewall filter is discarded.

You can plan and design stateless firewall filters to be used for various purposes—for example, to limit traffic to certain protocols, IP source or destination addresses, or data rates.

Class-of-Service Features

CoS features allow you to police and shape traffic.

- **Policing traffic**—Policers allow you to limit traffic of a certain class to a specified bandwidth and burst size. Packets exceeding the policer limits can be discarded or assigned to a different forwarding class, a different loss priority, or both. You can use policers to limit the amount of traffic passing into or out of an interface.
- **Traffic shaping**—You can shape traffic by assigning service levels with different delay, jitter, and packet loss characteristics to particular applications served by specific traffic flows. Traffic shaping is especially useful for real-time applications, such as voice and video transmission.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Session Characteristics for J Series Services Routers on page 73
- Understanding the Data Path for J Series Services Routers on page 79
- Monitoring Policy Statistics on page 132
- ALG Overview on page 169
- NAT Overview on page 927

Session Characteristics for J Series Services Routers

- Understanding Session Characteristics for J Series Services Routers on page 73
- Example: Controlling Session Termination for J Series Services Routers on page 74
- Example: Disabling TCP Packet Security Checks for J Series Services Routers on page 76
- Example: Accommodating End-to-End TCP Communication for J Series Services Routers on page 77

Understanding Session Characteristics for J Series Services Routers

Sessions are created, based on routing and other classification information, to store information and allocate resources for a flow. Sessions have characteristics, some of which you can change, such as when they are terminated. For example, you might want to ensure that a session table is never entirely full to protect against an attacker's attempt to flood the table and thereby prevent legitimate users from starting sessions.

Depending on the protocol and service, a session is programmed with a timeout value. For example, the default timeout for TCP is 1800 seconds. The default timeout for UDP is 60 seconds. When a flow is terminated, it is marked as invalid, and its timeout is reduced to 10 seconds.

If no traffic uses the session before the service timeout, the session is aged out and freed to a common resource pool for reuse. You can affect the life of a session in the following ways:

- You can specify circumstances for terminating sessions using any of the following methods:
 - Aggressively age out invalid sessions based on a timeout value
 - Age out sessions based on how full the session table is
 - Set an explicit timeout for aging out TCP sessions
 - Configure a TCP session to be invalidated when it receives a TCP RST (reset) message
- You can configure sessions to accommodate other systems as follows:
 - Disable TCP packet security checks
 - Accommodate end-to-end communication

The following topics show you how to modify a session's characteristics. For details, see the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Stateful and Stateless Data Processing for J Series Services Routers on page 69
 - Example: Controlling Session Termination for J Series Services Routers on page 74
 - Example: Disabling TCP Packet Security Checks for J Series Services Routers on page 76
 - Example: Accommodating End-to-End TCP Communication for J Series Services Routers on page 77

Example: Controlling Session Termination for J Series Services Routers

This example shows how to terminate sessions based on a timeout value or the number of sessions in the session table.

- Requirements on page 74
- Overview on page 75
- Configuration on page 75
- Verification on page 76

Requirements

Before you begin:

- Configure security zones. See “Security Zones and Interfaces Overview” on page 85.
- Configure security policies. See “Security Policies Configuration Overview” on page 120.

Overview

Junos OS terminates sessions normally under certain circumstances—for example, after receiving a TCP FINish Close or a RST (reset) message, when encountering ICMP errors for UDP, and when no matching traffic is received before the service timeout. When sessions are terminated, their resources are freed up for use by other sessions.

To control when sessions are terminated, you configure the router to age out sessions after a certain period of time, when the number of sessions in the session table reaches a specified percentage, or both. When the number of sessions in the session table reaches this percentage, the router begins to age sessions aggressively. When the number of sessions in the session table reaches the low-water mark, the router stops aggressively aging sessions.

Configuration

CLI Quick Configuration To quickly terminate sessions based on a timeout value or the number of sessions in the session table, copy the following commands and paste them into the CLI:

```
[edit]
set security flow aging early-ageout 2
set security flow aging high-watermark 90 low-watermark 50
set security flow tcp-session tcp-initial-timeout 280
set security flow tcp-session rst-invalidate-session
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To terminate sessions based on a timeout value or the number of sessions in the session table:

1. Specify the number of seconds after which a session is invalidated.

```
[edit security flow]
user@host# set aging early-ageout 2
```
2. Specify a percentage of sessions.

```
[edit security flow]
user@host# set aging high-watermark 90 low-watermark 50
```
3. Configure an explicit timeout value to remove a TCP session from the session table.

```
[edit security flow]
user@host# set tcp-session tcp-initial-timeout 280
```
4. Configure any session that receives a TCP RST message to be invalidated.

```
[edit security flow]
user@host# set tcp-session rst-initial-timeout 280
```

Results From configuration mode, confirm your configuration by entering the **show security flow** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security flow
aging {
    early-ageout 2;
    low-watermark 50;
    high-watermark 90;
}
tcp-session {
    rst-invalidate-session;
    tcp-initial-timeout 280;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 76

Troubleshooting with Logs

Purpose Use these logs to identify any issues.

Action From operational mode, enter the **show log messages** command and the **show log dcd** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Session Characteristics for J Series Services Routers on page 73
 - Example: Accommodating End-to-End TCP Communication for J Series Services Routers on page 77
 - Example: Disabling TCP Packet Security Checks for J Series Services Routers on page 76

Example: Disabling TCP Packet Security Checks for J Series Services Routers

This example shows how to disable TCP SYN checks and TCP sequence checks on all TCP sessions.

- Requirements on page 76
- Overview on page 76
- Configuration on page 77
- Verification on page 77

Requirements

Before you begin, review TCP packets and security checks. See *Junos OS Feature Support Reference for SRX Series and J Series Devices*.

Overview

Junos OS provides a mechanism to disable security checks on TCP packets to ensure interoperability with hosts and routers with faulty TCP implementations.

Configuration

Step-by-Step Procedure

To disable TCP SYN checks and TCP sequence checks on all TCP sessions:

1. Disable TCP SYN checks on all TCP sessions.

```
[edit security flow]
user@host# set tcp-session no-syn-check
```
2. Disable TCP sequence checks on all TCP sessions.

```
[edit security flow]
user@host# set tcp-session no-sequence-check
```
3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security flow** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Session Characteristics for J Series Services Routers on page 73
- Example: Controlling Session Termination for J Series Services Routers on page 74
- Example: Accommodating End-to-End TCP Communication for J Series Services Routers on page 77

Example: Accommodating End-to-End TCP Communication for J Series Services Routers

This example shows how to change the maximum segment size (MSS) for TCP packets to be sent or received over GRE and IPsec tunnels.

- Requirements on page 77
- Overview on page 77
- Configuration on page 78
- Verification on page 79

Requirements

Before you begin, review TCP packets and security checks. See *Junos OS Feature Support Reference for SRX Series and J Series Devices*.

Overview

End-to-end TCP communication in a customer network might not work for large packets approaching 1500 bytes because of GRE or IPsec tunneling encapsulation. You can configure sessions to accommodate other systems and segment sizes.

Configuration

CLI Quick Configuration To quickly change the MSS for TCP packets to be sent or received over GRE and IPsec tunnels, copy the following commands and paste them into the CLI:

```
[edit ]
set security flow tcp-mss ipsec-vpn mss 1400
set security flow tcp-mss gre-in mss 1364
set security flow tcp-mss gre-out mss 1364
set security flow tcp-mss all-tcp 1400
set security flow allow-dns-reply
set security flow route-change-timeout 62
set security flow syn-flood-protection-mode syn-proxy
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To change the MSS for TCP packets to be sent or received over GRE and IPsec tunnels:

1. Set the tunnel sessions.

```
[edit security flow]
user@host# set tcp-mss ipsec-vpn mss 1400
user@host# set tcp-mss gre-in mss 1364
user@host# set tcp-mss gre-out mss 1364
```

2. Configure TCP MSS for all TCP sessions.

```
[edit security flow]
user@host# set tcp-mss all-tcp 1400
```

3. Allow an unmatched incoming DNS reply packet.

```
[edit security flow]
user@host# set allow-dns-reply
```

4. Set the timeout value for route change to nonexistent route.

```
[edit security flow]
user@host# set route-change-timeout 62
```

5. Enable TCP SYN flood protection mode.

```
[edit security flow]
user@host# set syn-flood-protection-mode syn-proxy
```

Results From configuration mode, confirm your configuration by entering the **show security flow** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security flow
allow-dns-reply;
route-change-timeout 62;
syn-flood-protection-mode syn-proxy;
tcp-mss {
    all-tcp {
```

```
        mss 1400;
    }
    ipsec-vpn {
        mss 1400;
    }
    gre-in {
        mss 1364;
    }
    gre-out {
        mss 1364;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 79

Troubleshooting with Logs

Purpose Use these logs to identify any issues.

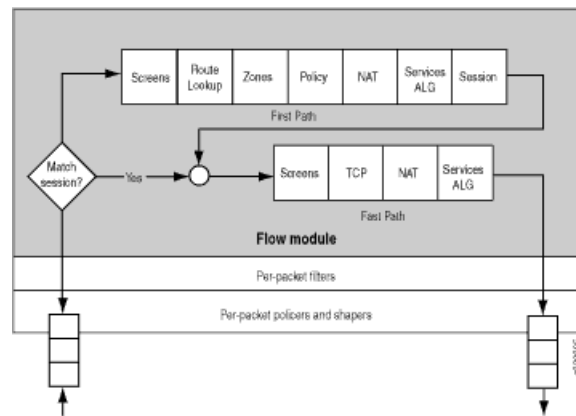
Action From operational mode, enter the **show log messages** command and the **show log dcd** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Example: Disabling TCP Packet Security Checks for J Series Services Routers on page 76
 - Example: Controlling Session Termination for J Series Services Routers on page 74
 - Understanding Session Characteristics for J Series Services Routers on page 73

Understanding the Data Path for J Series Services Routers

Figure 8 on page 80 shows the path of a data packet as it traverses the services router. Refer to Figure 6 on page 70 to see how the flow module in Figure 8 on page 80 fits in with the Junos operating system (Junos OS) architecture of the software.

Figure 8: Data Packet Traversing the Flow Module on the Services Router



As a packet transits the router, it takes the following path. This packet walk brings together the packet-based processing and flow-based processing that Junos OS performs on the packet.

- Understanding the Forwarding Processing on page 80
- Understanding the Session-Based Processing on page 80
- Understanding Forwarding Features on page 82

Understanding the Forwarding Processing

Junos OS performs forwarding processing as follows:

1. The packet enters the system and is treated on a per-packet basis.
2. The system applies stateless policing filters and class-of-service (CoS) classification to the packet.

For details, see the *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*, the *Junos OS Class of Service Configuration Guide for Security Devices*, and the *Junos OS CLI Reference*.

Understanding the Session-Based Processing

After forwarding processing, Junos OS performs session lookup and either first-packet processing or fast-path processing on the packet.

Session Lookup

If the packet has not already been dropped, Junos OS performs session lookup to determine whether the packet belongs to an existing session. The system uses six match criteria to perform the session lookup:

- Session token
- Source and destination IP addresses
- Source and destination ports
- Protocol

If the packet does not match an existing session, the system creates a new session for it. This process is called the first-packet path. (See “First-Packet Path Processing” on page 81.)

If the packet matches a session, fast-path processing is performed. (See “Fast-Path Processing” on page 82.)

First-Packet Path Processing

If a packet does not match an existing session, Junos OS creates a new session for it as follows:

1. For the first packet, the system creates a session based on the routing for the packet and the policy lookup so that the packet becomes the first packet of a flow.
2. Depending on the protocol and whether the service is TCP or UDP, the session is programmed with a timeout value.
 - For TCP, the default timeout is 1800 seconds.
 - For UDP, the default timeout is 60 seconds.

You can configure these timeouts to be more or less aggressive. If you have changed the session timeout value, the new value is applied here. If no traffic uses the session during the service timeout period, the router ages out the session and releases its memory for reuse.

3. Firewall screens are applied.

Session initialization screens are applied.
4. Route lookup is performed.
5. The destination zone is determined:
 - a. The system determines a packet's *incoming* zone by the interface through which it arrives.
 - b. The system determines a packet's *outgoing* zone by route lookup.

Together they determine which policy is applied to the packet.

6. Policy lookup is performed.

The system checks the packet against policies you have defined to determine how the packet is to be treated.
7. If Network Address Translation (NAT) is used, the system performs address allocation.
8. The system sets up the Application Layer Gateway (ALG) service vector.
9. The system creates and installs the session.

Decisions made for the first packet of a flow are cached in a flow table for use with following related flows.

For example, the system determines asymmetric traffic by doing a reverse route lookup on the packet. If the first packet of a flow has ingress on an interface for a

zone, then the reply traffic for this flow needs to egress out of the same interface on which the first packet ingress; otherwise, the traffic is considered asymmetric and will be dropped.

10. Fast-path processing is applied to the packet.

Fast-Path Processing

If a packet matches a session, Junos OS performs fast-path processing as follows:

1. Configured screens are applied.
2. TCP checks are performed.
3. NAT is applied.
4. Forwarding features are applied. See “Understanding Forwarding Features” on page 82.

Understanding Forwarding Features

After the packet has passed through session-based processing, Junos OS prepares the packet and transmits it as follows:

1. Routing packet filters are applied.
2. Traffic shaping is applied.
3. The packet is transmitted.

For information about packet filters and CoS traffic shaping, see the *Junos OS Class of Service Configuration Guide for Security Devices*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Stateful and Stateless Data Processing for J Series Services Routers on page 69
 - NAT Overview on page 927
 - Security Policies Overview on page 115
 - ALG Overview on page 169

PART 2

Security Zones and Interfaces

- Security Zones and Interfaces on page 85
- Address Books and Address Sets on page 103

CHAPTER 4

Security Zones and Interfaces

- Security Zones and Interfaces Overview on page 85
- Security Zones on page 86
- Host Inbound Traffic on page 90
- Protocols on page 95
- TCP-Reset Parameters on page 97
- DNS on page 99

Security Zones and Interfaces Overview

Interfaces act as a doorway through which traffic enters and exits a Juniper Networks device. Many interfaces can share exactly the same security requirements; however, different interfaces can also have different security requirements for inbound and outbound data packets. Interfaces with identical security requirements can be grouped together into a single *security zone*.

A *security zone* is a collection of one or more network segments requiring the regulation of inbound and outbound traffic through policies.

Security zones are logical entities to which one or more interfaces are bound. With many types of Juniper Networks devices, you can define multiple security zones, the exact number of which you determine based on your network needs.

On a single device, you can configure multiple security zones, dividing the network into segments to which you can apply various security options to satisfy the needs of each segment. At a minimum, you must define two security zones, basically to protect one area of the network from the other. On some security platforms, you can define many security zones, bringing finer granularity to your network security design—and without deploying multiple security appliances to do so.

From the perspective of security policies, traffic enters into one security zone and goes out on another security zone. This combination of a **from-zone** and a **to-zone** is defined as a *context*. Each context contains an ordered list of policies. For more information on policies, see “Security Policies Overview” on page 115.

This topic includes the following sections:

- Understanding Security Zone Interfaces on page 86
- Understanding Interface Ports on page 86

Understanding Security Zone Interfaces

An interface for a security zone can be thought of as a doorway through which TCP/IP traffic can pass between that zone and any other zone.

Through the policies you define, you can permit traffic between zones to flow in one direction or in both. With the routes that you define, you specify the interfaces that traffic from one zone to another must use. Because you can bind multiple interfaces to a zone, the routes you chart are important for directing traffic to the interfaces of your choice.

An interface can be configured with an IPv4 address, IPv6 address, or both.

Understanding Interface Ports

On J Series Services Routers, interface ports for the system are located on Physical Interface Modules (PIMs) that you can install in slots on the device. In addition, each device has four built-in Gigabit Ethernet ports in slot 0. Each physical port can have many logical interfaces configured with properties different from the port's other logical units.

Interfaces are named by type, slot number, module number (always 0), port number, and the logical unit number. Port numbering starts with 0. Interface names have the following format:

`type-pim/0/port.logical-unit-number`

For example, an interface on port 1 of a T1 PIM installed in slot 3 is named `t1-3/0/1`. Logical unit 1 on the interface is named `t1-3/0/1.1`. The built-in Gigabit Ethernet interfaces are named `ge-0/0/0` through `ge-0/0/3`.

For more information about interfaces and interface names, see the *Junos OS Interfaces Configuration Guide for Security Devices*.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Functional Zones on page 87
- Understanding Security Zones on page 87
- Example: Creating Security Zones on page 88
- Understanding How to Control Inbound Traffic Based on Traffic Types on page 90

Security Zones

- Understanding Functional Zones on page 87
- Understanding Security Zones on page 87
- Example: Creating Security Zones on page 88

Understanding Functional Zones

A functional zone is used for special purposes, like management interfaces. Currently, only the management (MGT) zone is supported. Management zones have the following properties:

- Management zones host management interfaces.
- Traffic entering management zones does not match policies; therefore, traffic cannot transit out of any other interface if it was received in the management interface.
- Management zones can only be used for dedicated management interfaces.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Zones and Interfaces Overview on page 85
 - Understanding Security Zones on page 87
 - Example: Creating Security Zones on page 88

Understanding Security Zones

Security zones are the building blocks for policies; they are logical entities to which one or more interfaces are bound. Security zones provide a means of distinguishing groups of hosts (user systems and other hosts, such as servers) and their resources from one another in order to apply different security measures to them.

Security zones have the following properties:

- **Policies**—Active security policies that enforce rules for the transit traffic, in terms of what traffic can pass through the firewall, and the actions that need to take place on the traffic as it passes through the firewall. For more information, see “Security Policies Overview” on page 115.
- **Screens**—A Juniper Networks stateful firewall secures a network by inspecting, and then allowing or denying, all connection attempts that require passage from one security zone to another. For every security zone, you can enable a set of predefined screen options that detect and block various kinds of traffic that the device determines as potentially harmful. For more information, see “Reconnaissance Deterrence Overview” on page 711.
- **Address books**—IP addresses and address sets that make up an address book to identify its members so that you can apply policies to them. Address book entries can include any combination of IPv4 addresses, IPv6 addresses, and Domain Name System (DNS) names. For more information, see “Example: Configuring Address Books” on page 108.
- **TCP-RST**—When this feature is enabled, the system sends a TCP segment with the RESET flag set when traffic arrives that does not match an existing session and does not have the SYNchronize flag set.
- **Interfaces**—List of interfaces in the zone.

Security zones have the following preconfigured zones:

- junos-global zone—Defined in the Junos OS defaults and cannot be configured by the user. The global zone serves as a storage area for static NAT addresses and can be used in policies like any other security zone.
- Trust zone—Available only in the factory configuration and is used for initial connection to the device. After you commit a configuration, the trust zone can be overridden.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Zones and Interfaces Overview on page 85
 - Understanding Functional Zones on page 87
 - Example: Creating Security Zones on page 88

Example: Creating Security Zones

This example shows how to configure zones and assign interfaces to them. When you configure a security zone, you can specify many of its parameters at the same time.

- Requirements on page 88
- Overview on page 88
- Configuration on page 88
- Verification on page 89

Requirements

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

Overview

An interface for a security zone can be thought of as a doorway through which TCP/IP traffic can pass between that zone and any other zone.



NOTE: By default, interfaces are in the null zone. The interfaces will not pass traffic until they have been assigned to a zone.

Configuration

CLI Quick Configuration

To quickly create zones and assign interfaces to them, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/1 unit 0 family inet address 10.12.12.1/24
set interfaces ge-0/0/1 unit 0 family inet6 address fa:43::21/96
set security security-zone ABC interfaces ge-0/0/1.0
```


Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To create zones and assign interfaces to them:

1. Configure an Ethernet interface and assign an IPv4 address to it.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.12.12.1/24
```
2. Configure an Ethernet interface and assign an IPv6 address to it.

```
user@host# set interfaces ge-0/0/1 unit 0 family inet6 address fa:43::21/96
```
3. Configure a security zone and assign it to an Ethernet interface.

```
user@host# set security security-zone ABC interfaces ge-0/0/1.0
```

Results From configuration mode, confirm your configuration by entering the **show security zones security-zone ABC** and **show interfaces ge-0/0/1** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show security zones security-zone ABC
...
  interfaces {
    ge-0/0/1.0 {
      ...
    }
  }
[edit]
user@host# show interfaces ge-0/0/1
...
  unit 0 {
    family inet {
      address 10.12.12.1/24;
    }
    family inet6 {
      address fe:43::21/96;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 89

Troubleshooting with Logs

Purpose Use these logs to identify any issues.

Action From operational mode, enter the **show log messages** command and the **show log dcd** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Zones and Interfaces Overview on page 85
 - Understanding Functional Zones on page 87
 - Understanding Security Zones on page 87

Host Inbound Traffic

- Understanding How to Control Inbound Traffic Based on Traffic Types on page 90
- Supported System Services for Host Inbound Traffic on page 91
- Example: Controlling Inbound Traffic Based on Traffic Types on page 92

Understanding How to Control Inbound Traffic Based on Traffic Types

This topic describes how to configure zones to specify the kinds of traffic that can reach the device from systems that are directly connected to its interfaces.

Note the following:

- You can configure these parameters at the zone level, in which case they affect all interfaces of the zone, or at the interface level. (Interface configuration overrides that of the zone.)
- You must enable all expected host-inbound traffic. Inbound traffic destined to this device is dropped by default.
- You can also configure a zone's interfaces to allow for use by dynamic routing protocols.

This feature allows you to protect the device against attacks launched from systems that are directly or indirectly connected to any of its interfaces. It also enables you to selectively configure the device so that administrators can manage it using certain applications on certain interfaces. You can prohibit use of other applications on the same or different interfaces of a zone. For example, most likely you would want to ensure that outsiders not use the Telnet application from the Internet to log into the device because you would not want them connecting to your system.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Zones and Interfaces Overview on page 85
 - Supported System Services for Host Inbound Traffic on page 91
 - Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter on page 98
 - Example: Controlling Inbound Traffic Based on Traffic Types on page 92

Supported System Services for Host Inbound Traffic

This topic describes the supported system services for host inbound traffic on the specified zone or interface.

For example, suppose a user whose system was connected to interface **1.3.1.4** in zone **ABC** wanted to telnet into interface **2.1.2.4** in zone **ABC**. For this action to be allowed, the Telnet application must be configured as an allowed inbound service on both interfaces and a policy must permit the traffic transmission.

Table 10 on page 91 shows the system services that can be used for host inbound traffic.

Table 10: System Services for Host Inbound Traffic

Host Inbound System Services	
all	any-service
dns	finger
ftp	http
https	indent-reset
ike	netconf
ntp	ping
reverse-ssh	reverse-telnet
rlogin	rpm
rsh	sip
snmp	snmp-trap
ssh	telnet
tftp	traceroute
xnm-clear-text	xnm-ssl

Table 11 on page 91 shows the supported protocols that can be used for host inbound traffic.

Table 11: Protocols for Host Inbound Traffic

Protocols	
all	bfd

Table 11: Protocols for Host Inbound Traffic (*continued*)

Protocols	
bgp	dvmrp
igmp	msdp
ndp	nhrp
ospf	ospf3
pgm	pim
rip	ripng
sap	vrrp



NOTE: All services (except DHCP and BOOTP) can be configured either per zone or per interface. A DHCP server is configured only per interface because the incoming interface must be known by the server to be able to send out DHCP replies.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding How to Control Inbound Traffic Based on Traffic Types on page 90
- Example: Controlling Inbound Traffic Based on Traffic Types on page 92

Example: Controlling Inbound Traffic Based on Traffic Types

This example shows how to configure inbound traffic based on traffic types.

- Requirements on page 92
- Overview on page 92
- Configuration on page 93
- Verification on page 94

Requirements

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Understand Inbound traffic types. See “Understanding How to Control Inbound Traffic Based on Traffic Types” on page 90.

Overview

By allowing system services to run, you can configure zones to specify different types of traffic that can reach the device from systems that are directly connected to its interfaces.

You can configure the different system services at the zone level, in which case they affect all interfaces of the zone, or at the interface level. (Interface configuration overrides that of the zone.)

You must enable all expected host-inbound traffic. Inbound traffic from devices directly connected to the device's interfaces is dropped by default.

Configuration

CLI Quick Configuration To quickly configure inbound traffic based on traffic types, copy the following commands and paste them into the CLI:

```
[edit]
set security zones security-zone ABC host-inbound-traffic system-services all
set security zones security-zone ABC interfaces ge-0/0/1.3 host-inbound-traffic
  system-services telnet
set security zones security-zone ABC interfaces ge-0/0/1.3 host-inbound-traffic
  system-services ftp
set security zones security-zone ABC interfaces ge-0/0/1.3 host-inbound-traffic
  system-services snmp
set security zones security-zone ABC interfaces ge-0/0/1.0 host-inbound-traffic
  system-services all
set security zones security-zone ABC interfaces ge-0/0/1.0 host-inbound-traffic
  system-services ftp except
set security zones security-zone ABC interfaces ge-0/0/1.0 host-inbound-traffic
  system-services http except
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure inbound traffic based on traffic types:

1. Configure a security zone.

```
[edit]
user@host# edit security zones security-zone ABC
```
2. Configure the security zone to support inbound traffic for all system services.

```
[edit security zones security-zone ABC]
user@host# set host-inbound-traffic system-services all
```
3. Configure the Telnet, FTP, and SNMP system services at the interface level (not the zone level) for the first interface.

```
[edit security zones security-zone ABC]
user@host# set interfaces ge-0/0/1.3 host-inbound-traffic system-services telnet
user@host# set interfaces ge-0/0/1.3 host-inbound-traffic system-services ftp
user@host# set interfaces ge-0/0/1.3 host-inbound-traffic system-services snmp
```
4. Configure the security zone to support inbound traffic for all system services for a second interface.

```
[edit security zones security-zone ABC]
user@host# set interfaces ge-0/0/1.0 host-inbound-traffic system-services all
```

5. Exclude the FTP and HTTP system services from the second interface.

```
[edit security zones security-zone ABC]
user@host# set interfaces ge-0/0/1.0 host-inbound-traffic system-services ftp
except
user@host# set interfaces ge-0/0/1.0 host-inbound-traffic system-services http
except
```

Results From configuration mode, confirm your configuration by entering the **show security zones security-zone ABC**. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security zones security-zone ABC
host-inbound-traffic {
    system-services {
        all;
    }
}
interfaces {
    ge-0/0/1.3 {
        host-inbound-traffic {
            system-services {
                ftp;
                telnet;
                snmp;
            }
        }
    }
    ge-0/0/1.0 {
        host-inbound-traffic {
            system-services {
                all;
                ftp {
                    except;
                }
                http {
                    except;
                }
            }
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 94

Troubleshooting with Logs

Purpose Use these logs to identify any issues.

Action From operational mode, enter the **show log messages** command and the **show log dcd** command.

- Related Topics
- *Junos OS CLI Reference*
 - *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding How to Control Inbound Traffic Based on Traffic Types on page 90
 - Supported System Services for Host Inbound Traffic on page 91

Protocols

- Understanding How to Control Inbound Traffic Based on Protocols on page 95
- Example: Controlling Inbound Traffic Based on Protocols on page 96

Understanding How to Control Inbound Traffic Based on Protocols

This topic describes the inbound system protocols on the specified zone or interface.

Any host-inbound traffic that corresponds to a protocol listed under the host-inbound traffic option is allowed. For example, if anywhere in the configuration, you map a protocol to a port number other than the default, you can specify the protocol in the host-inbound traffic option, and the new port number will be used. Table 12 on page 95 lists the supported protocols. A value of **all** indicates that traffic from all of the following protocols is allowed inbound on the specified interfaces (of the zone, or a single specified interface).

Table 12: Supported Inbound System Protocols

Supported System Services			
all	igmp	pim	sap
bfd	ldp	rip	vrrp
bgp	msdp	ripng	nhrp
router-discovery	dvmrp	ospf	rsvp
ndp	pgm	ospf3	



NOTE: If DVMRP or PIM is enabled for an interface, IGMP and MLD host-inbound traffic is enabled automatically. Because ISIS uses OSI addressing and should not generate any IP traffic, there is no host-inbound traffic option for the ISIS protocol.

- Related Topics
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Zones and Interfaces Overview on page 85
 - Understanding How to Control Inbound Traffic Based on Traffic Types on page 90
 - Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter on page 98

- Example: Controlling Inbound Traffic Based on Protocols on page 96

Example: Controlling Inbound Traffic Based on Protocols

This example shows how to enable inbound traffic for an interface.

- Requirements on page 96
- Overview on page 96
- Configuration on page 96
- Verification on page 97

Requirements

Before you begin:

- Configure security zones. See “Example: Creating Security Zones” on page 88.
- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

Overview

Any host-inbound traffic that corresponds to a protocol listed under the host-inbound traffic option is allowed. For example, if anywhere in the configuration you map a protocol to a port number other than the default, you can specify the protocol in the host-inbound traffic option, and the new port number will be used.

A value of **all** indicates that traffic from all of the protocols is allowed inbound on the specified interfaces (of the zone, or a single specified interface).

Configuration

CLI Quick Configuration

To quickly configure inbound traffic based on protocols, copy the following commands and paste them into the CLI:

```
[edit]
set security zones security-zone ABC interfaces ge-0/0/1.0 host-inbound-traffic protocols
  ospf
set security zones security-zone ABC interfaces ge-0/0/1.0 host-inbound-traffic protocols
  ospf3
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure inbound traffic based on protocols:

1. Configure a security zone.

```
[edit]
user@host# edit security zones security-zone ABC
```

2. Configure the security zone to support inbound traffic based on the ospf protocol for an interface.


```
[edit security zones security-zone ABC]
user@host# set interfaces ge-0/0/1.0 host-inbound-traffic protocols ospf
```

3. Configure the security zone to support inbound traffic based on the ospf3 protocol for an interface.

```
[edit security zones security-zone ABC]
user@host# set interfaces ge-0/0/1.0 host-inbound-traffic protocols ospf3
```

Results From configuration mode, confirm your configuration by entering the **show security zones security-zone ABC**. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security zones security-zone ABC
interfaces {
  ge-0/0/1.0 {
    host-inbound-traffic {
      protocols {
        ospf;
        ospf3;
      }
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 97

Troubleshooting with Logs

Purpose Use these logs to identify any issues.

Action From operational mode, enter the **show log messages** command and the **show log dcd** command.

- Related Topics**
- *Junos OS CLI Reference*
 - *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding How to Control Inbound Traffic Based on Protocols on page 95

TCP-Reset Parameters

- Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter on page 98
- Example: Configuring the TCP-Reset Parameter on page 98

Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter

When the TCP-RST feature is enabled, the system sends a TCP segment with the RESET flag set when traffic arrives that does not match an existing session and does not have the SYNchronize flag set.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Zones and Interfaces Overview on page 85
 - Understanding How to Control Inbound Traffic Based on Traffic Types on page 90
 - Understanding How to Control Inbound Traffic Based on Protocols on page 95
 - Example: Configuring the TCP-Reset Parameter on page 98

Example: Configuring the TCP-Reset Parameter

This example shows how to configure the TCP-Reset parameter for a zone.

- Requirements on page 98
- Overview on page 98
- Configuration on page 98
- Verification on page 99

Requirements

Before you begin, configure security zones. See “Example: Creating Security Zones” on page 88.

Overview

When the TCP-Reset parameter feature is enabled, the system sends a TCP segment with the RESET flag set when traffic arrives that does not match an existing session and does not have the SYNchronize flag set.

Configuration

Step-by-Step Procedure To configure the TCP-Reset parameter for a zone:

1. Configure a security zone.
`[edit]
user@host# edit security zones security-zone ABC`
2. Configure the TCP-Reset parameter for the zone.
`[edit security zones security-zone ABC]
user@host# set tcp-rst`
3. If you are done configuring the device, commit the configuration.
`[edit]
user@host# commit`

Verification

To verify the configuration is working properly, enter the **show security zones** command.

Related Topics

- *Junos OS CLI Reference*
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding How to Identify Duplicate Sessions Using the TCP-Reset Parameter on page 98

DNS

- DNS Overview on page 99
- Example: Configuring the TTL Value for DNS name servers on page 100
- Example: Configuring a Forwarder for a DNS server on page 100
- DNSSEC Overview on page 100
- Example: Configuring DNSSEC on page 101
- Example: Configuring Keys for DNSSEC on page 101
- Example: Configuring Secure Domains and Trusted Keys for DNSSEC on page 102

DNS Overview

A Domain Name System (DNS) is a distributed hierarchical system that converts hostnames to IP addresses. The DNS is divided into sections called zones. Each zone has name servers that respond to the queries belonging to their zones.

This topic includes the following sections:

- DNS Components on page 99
- DNS Server Caching on page 99
- Forwarders on page 100

DNS Components

DNS includes three main components:

- DNS resolver — Resides on the client side of the DNS. When a user sends a hostname request, the resolver sends a DNS query request to the name servers to request the hostname's IP address.
- Name servers — Processes the DNS query requests received from the DNS resolver and returns the IP address to the resolver.
- Resource records — Data elements that define the basic structure and content of the DNS.

DNS Server Caching

DNS name servers are responsible for providing the hostname IP address to users. The TTL field in the resource record defines the period for which DNS query results are cached.

When the TTL value expires, the name server sends a fresh DNS query and updates the cache.

Forwarders

When a DNS server cannot resolve a query, it forwards the query to another DNS server that is configured as a forwarder. You can use the CLI to configure a DNS server to act as a forwarder. The DNS server forwards the queries only to the servers that are configured as forwarders.

- Related Topics**
- Example: Configuring the TTL Value for DNS name servers on page 100
 - Example: Configuring a Forwarder for a DNS server on page 100
 - DNSSEC Overview on page 100

Example: Configuring the TTL Value for DNS name servers

The DNS server stores DNS query responses in its cache for the TTL period specified in the TTL field of the resource record. You can configure the TTL value for cached responses by using the CLI. The following example shows how you configure the TTL value for a DNS server cache:

```
[edit]
user@host# set system services dns max-cache-ttl ttl-value
```

The configurable range varies from 0 to 604800 seconds.

You can also configure the TTL value for cached negative responses by using the CLI:

```
[edit]
user@host# set system services dns max-ncache-ttl ttl-value
```

- Related Topics**
- DNS Overview on page 99

Example: Configuring a Forwarder for a DNS server

You can configure a DNS server to act as a forwarder. A DNS server will forward any DNS query it cannot handle to another server that is configured as a forwarder. The following example shows how to configure a DNS server with IP address 10.100.11.24 to act as a forwarder:

```
edit
user@host# set system services dns forwarders 10.100.11.24
```

- Related Topics**
- DNS Overview on page 99

DNSSEC Overview

Junos OS devices support the domain name service security extensions (DNSSEC) standard. DNSSEC is an extension of DNS that provides authentication and integrity verification of data by using public-key based signatures.

In DNSSEC, all the resource records in a DNS are signed with the private key of the zone owner. The DNS resolver uses the public key of the owner to validate the signature. The zone owner generates a private key to encrypt the hash of a set of resource records. The private key is stored in RRSIG record. The corresponding public key is stored in the DNSKEY record. The resolver uses the public key to decrypt the RRSIG and compares the result with the hash of the resource record to verify that it has not been altered.

Similarly, the hash of the public DNSKEY is stored in a DS record in a parent zone. The zone owner generates a private key to encrypt the hash of the public key. The private key is stored in the RRSIG record. The resolver retrieves the DS record and its corresponding RRSIG record and public key. Using the public key, the resolver decrypts the RRSIG record and compares the result with the hash of the public DNSKEY to verify that it has not been altered. This establishes a chain of trust between the resolver and the name servers.

- Related Topics**
- DNS Overview on page 99
 - Example: Configuring Keys for DNSSEC on page 101
 - Example: Configuring Secure Domains and Trusted Keys for DNSSEC on page 102

Example: Configuring DNSSEC

DNS-enabled devices run a DNS resolver (proxy) that listens on loopback address 127.0.0.1 or ::1. The DNS resolver performs a hostname resolution for DNSSEC. Users need to set name server IP address to 127.0.0.1 or ::1 so the DNS resolver forwards all DNS queries to DNSSEC instead of to DNS. If the name server IP address is not set, DNS will handle all queries instead of to DNSSEC.

The following example shows how to set the server IP address to 127.0.0.1:

```
[edit]
user@host# set system name-server 127.0.0.1
```

The DNSSEC feature is enabled by default. You can disable DNSSEC in the server by using the following CLI command:

```
[edit ]
set system services dns dnssec disable
```

- Related Topics**
- DNSSEC Overview on page 100

Example: Configuring Keys for DNSSEC

You can load a public key from a file or you can copy and paste the key file from a terminal. In both cases, you must save the keys to the configuration instead of to a file. The following example shows how to load a key from a file:

```
[edit system services dns dnssec trusted-keys]
#load-key filename
```

The following example explains how to load the key from a terminal:

```
[edit system services dns dnssec trusted-keys]
# set key "...pasted-text..."
```

If you are done loading the keys from the file or terminal, click **commit** in the CLI editor.

Related Topics • Example: Configuring Secure Domains and Trusted Keys for DNSSEC on page 102

Example: Configuring Secure Domains and Trusted Keys for DNSSEC

Users can configure secure domains and assign trusted keys to the domains by using CLI commands. Both signed and unsigned responses can be validated when DNSSEC is enabled. The following example shows how to configure domain1.net and domain2.net as secure domains:

```
[edit]
user@host# set system services dns dnssec secure-domain domain1.net
user@host# set system services dns dnssec secure-domain domain2.net
```

When you configure a domain as a secure domain and if DNSSEC is enabled, all unsigned responses to that domain are ignored and the server returns a SERVFAIL error code to the client for the unsigned responses. If the domain is not configured as a secure domain, unsigned responses will be accepted.

The following example shows how to configure trusted keys to domain1.net:

```
[edit]
user@host# set system services dns dnssec secure-domain domain1.net trusted-keys
key "domain1.net 256 3 3 \"CJ+tJ5...\""; key "dlv.isc.org.256 3 3 \"CPIfHBL...\""
```

When the server receives a signed response, it checks if the DNSKEY in the response matches any of the trusted keys configured above. If it finds a match, the server accepts the signed response.

You can also attach a trusted anchor to a secure domain to validate the signed responses. The following example shows how to attach a root zone dlv.isc.org as a trusted anchor to domain2.net:

```
[edit]
user@host# set system services dns dnssec dlv domain domain2.net trusted-anchor
dlv.isc.org
```

When the server receives a signed response, it queries the dlv.isc.org for a DS record. When it receives the DS record, it checks if the DNSKEY in the DS record matches the DNSKEY in the signed response. If it finds a match, the server accepts the signed response.

Related Topics • DNSSEC Overview on page 100
• Example: Configuring Keys for DNSSEC on page 101

CHAPTER 5

Address Books and Address Sets

- Security Policy Address Books and Address Sets Overview on page 103
- Understanding Address Books on page 104
- Understanding Address Sets on page 105
- Limitations of Addresses and Address Sets on page 107
- Example: Configuring Address Books on page 108
- Verifying Address Book Configuration on page 110

Security Policy Address Books and Address Sets Overview

Each security zone contains an address book. Before you can set up policies between two zones, you must define the addresses for each of the zone's address books. To manage an address book with large numbers of addresses, you can create groups of addresses called address sets.

A security zone is a logical group of interfaces with identical security requirements. Each security zone contains an address book. Before you can set up policies between two zones, you must define the addresses for each of the zone's address books. A zone's address book must contain entries for the addressable networks and end hosts (and, thus, users) belonging to the zone.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Security Policies Overview on page 115
- Understanding Address Books on page 104
- Understanding Address Sets on page 105
- Example: Configuring Address Books on page 108
- Verifying Address Book Configuration on page 110

Understanding Address Books

The following guidelines apply to address books:

- An address book for a security zone contains the IP address or domain names of hosts and subnets whose traffic is either allowed, blocked, encrypted, or user-authenticated.
- Address books can have address sets. Each address set has a name and a list of address names.
- Addresses and address sets in the same zone must have distinct names.
- Addresses must conform to the security requirements of the zone.
- Address book entries can include any combination of IPv4 addresses, IPv6 addresses, and Domain Name System (DNS) names.
- The predefined address **any** is automatically created for each security zone.
- The address book of a security zone must contain all IP addresses that are reachable within that zone.

Policies contain both source and destination zones and addresses. An address is referred to in a policy by the name you give it in the zone address book.

- When traffic is sent to a zone, the zone and address to which the traffic is sent are used as the destination zone and address-matching criteria in policies.
- When traffic is sent from a zone, the zone and address from which it is sent are used as the matching source zone and address in policies.

For more information on the address book configuration syntax and options, see the *Junos OS CLI Reference*.



NOTE: Specify addresses as network prefixes in the *prefix/length* format. For example, 1.2.3.0/24 is an acceptable address book address because it translates to a network prefix. However, 1.2.3.4/24 is not acceptable for an address book because it exceeds the subnet length of 24 bits. Everything beyond the subnet length must be entered as 0 (zero). In special scenarios, you can enter a hostname because it can use the full 32-bit address length.

An IPv6 address prefix is a combination of an IPv6 prefix (address) and a prefix length. The prefix takes the form *ipv6-prefix/prefix-length* and represents a block of address space (or a network). The *ipv6-prefix* variable follows general IPv6 addressing rules. The */prefix-length* variable is a decimal value that indicates the number of contiguous, higher-order bits of the address that make up the network portion of the address. For example, 10FA:6604:8136:6502::/64 is a possible IPv6 prefix. For more information on text representation of IPv6 addresses and address prefixes, see RFC 4291, *IP Version 6 Addressing Architecture*.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Security Policy Address Books and Address Sets Overview on page 103
- Understanding Address Sets on page 105
- Example: Configuring Address Books on page 108
- Example: Configuring Schedulers (CLI) on page 136

Understanding Address Sets

An address book can grow to contain large numbers of addresses and become difficult to manage. To manage an address book with large numbers of addresses, you can create groups of addresses called *address sets*. You can reference an address set in a policy as you would an individual address book entry.

The following example shows addresses and address sets in the green zone:

```

user@host# set security zones security-zone green address-book address src_addr1
64.10.4.44/32
user@host# set security zones security-zone green address-book address src_addr2
64.10.9.28/32
user@host# set security zones security-zone green address-book address src_addr3
10.10.10.0/24
user@host# set security zones security-zone green address-book address src_addr4
fa:43::/96
user@host# set security zones security-zone green address-book address src_addr5
fe80::210:dbff:feff:1000/64
user@host# set security zones security-zone green address-book address src_addr6
0001:db8:1::1/127
user@host# set security zones security-zone green address-book address bbc dns-name
www.bbc.com
user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr1
user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr2
user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr3
user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr4
user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr5
user@host# set security zones security-zone green address-book address-set
my_source_addresses address src_addr6
user@host# show security zones security-zone green

address-book {
    address src_addr1 64.10.4.44/32;
    address src_addr2 64.10.9.28/32;
    address src_addr3 10.10.10.0/24;
    address src_addr4 fa:43::/96;
    address src_addr5 fe80::210:dbff:feff:1000/64;
    address src_addr6 0001:db8:1::1/127;
    address bbc {
        dns-name www.bbc.com;
    }
    address-set my_source_addresses {
        address src_addr1;
        address src_addr2;
        address src_addr3;
    }
}

```

```

        address src_addr4;
        address src_addr5;
        address src_addr6;
    }
}

```

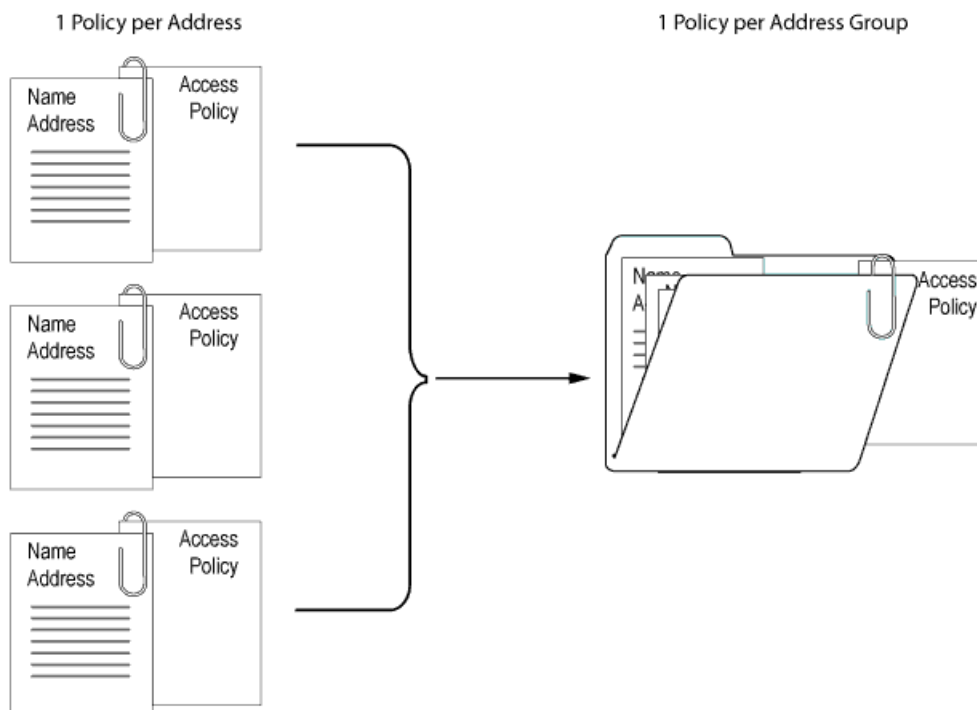
For more information on the address set configuration syntax and options, see the *Junos OS CLI Reference*.



NOTE: Consider that for each address set, the system creates individual rules for its members. It creates an internal rule for each member in the group as well as for each service configured for each user. If you configure address books without taking this into account, you can exceed the number of available policy resources, especially if both the source and destination addresses are address groups and the specified service is a service group.

When you add addresses to policies, sometimes the same subset of addresses can be present in multiple policies, making it difficult to manage how policies affect each address entry. Junos OS allows you to create groups of addresses called *address sets*. Address sets simplify the process by allowing you to add multiple addresses within an address set and therefore manage a small number of address sets, rather than manage a large number of individual address entries. See Figure 9 on page 106.

Figure 9: Address Sets



The address set option has the following features:

- You can create address sets in any zone.
- You can create address sets with existing users, or you can create empty address sets and later fill them with users.
- You can reference an address set entry in a policy like an individual address book entry.



NOTE: Junos OS applies policies automatically to each address set member, so you do not have to create them one by one for each address.

- When you delete an individual address book entry from the address book, you must remove the address (wherever it is referred) from all the address sets.

The following constraints apply to address sets:

- To configure an address set, you need more than an address in the address book.
- Address sets can only contain address names that belong to the same security zone.
- Address names cannot be the same as address set names. For example, if the name **Paris** is used for an address in an individual address entry, it cannot be used for an address set name.
- If an address set is referenced in a policy, the address set cannot be removed without removing its reference in the policy. It can, however, be edited.
- You cannot add the predefined address **any** to an address book.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Security Policy Address Books and Address Sets Overview on page 103
- Understanding Address Books on page 104
- Example: Configuring Address Books on page 108
- Example: Configuring Schedulers (CLI) on page 136
- Limitations of Addresses and Address Sets on page 107

Limitations of Addresses and Address Sets

On SRX Series and J Series devices, the limitation on the number of addresses in address-set has been increased. The number of addresses in address-set now depends on the device and is equal to the number of addresses supported by the policy.

Table 13 on page 108 provides the address-set details per device to increase the configuration limitation.

Table 13: Number of Addresses in address-set on SRX Series and J Series Devices

Device	address-set
Default	1024
SRX100 High Memory	1024
SRX100 Low Memory	512
SRX210 High Memory	1024
SRX210 Low Memory	512
SRX240 High Memory	1024
SRX240 Low Memory	512
SRX650	1024
SRX3400	1024
SRX3600	1024
SRX5600	1024
SRX5800	1024
J Series	1024

Example: Configuring Address Books

This example describes how to configure address books and address sets for a zone.

- Requirements on page 108
- Overview on page 108
- Configuration on page 109
- Verification on page 110

Requirements

Before you begin, configure the zones required in this example. See “Example: Creating Security Zones” on page 88.

Overview

In this example, you configure addresses and address sets for address books in the IntranetGREEN zone. This zone contains servers that belong to the same subnet. You can add individual addresses for the servers to the zone address list to accommodate

users with access rights to one server but not the other. You can also add an address set to combine the servers into a single addressable entity.

Configuration

CLI Quick Configuration To quickly configure address book entries for the IntranetGREEN zone, copy the following commands and paste them into the CLI.

```
[edit]
set security zones security-zone IntranetGREEN address-book address G1 10.1.10.0/24
set security zones security-zone IntranetGREEN address-book address G2 192.168.0.0/16
set security zones security-zone IntranetGREEN address-book address-set SerAll address
  G1
set security zones security-zone IntranetGREEN address-book address-set SerAll address
  G2
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure address book entries:

1. Create a security zone.

```
[edit]
user@host# set security zones security-zone IntranetGREEN
```

2. Create an address book and assign an address entry.

```
[edit security zones security-zone IntranetGREEN]
user@host# set address-book address G1 10.1.10.0/24
```

3. Create another address book and assign an address entry.

```
[edit security zones security-zone IntranetGREEN]
user@host# set address-book address G2 192.168.0.0/16
```

4. Configure an address set for all of the entries in Step 2.

```
[edit security zones security-zone IntranetGREEN]
user@host# set address-book address-set serAll address G1
```

5. Configure another address set for the entries in Step 3.

```
[edit security zones security-zone IntranetGREEN]
user@host# set address-book address-set serAll address G2
```

Results From configuration mode, confirm your configuration by entering the **show security zones security-zone IntranetGREEN** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
[user@host]show security zones security-zone IntranetGREEN
address-book {
  address G1 10.1.10.0/24;
  address G2 192.168.0.0/16;
  address-set serAll {
```

```
        address G1;
        address G2;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Address book Entries on page 110

Verifying the Address book Entries

Purpose Verify the list of address book entries currently configured in the device.

Action From operational mode, enter the **show security zones** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Address Books and Address Sets Overview on page 103
 - Verifying Address Book Configuration on page 110

Verifying Address Book Configuration

Purpose Display information about address books and zones.

Action Use the **show security zones** CLI command to verify the address book and address set configuration. You get the following output:

```
user@host# show security zones security-zone green
```

```
address-book {
  address src_addr1 64.10.4.44/32;
  address src_addr2 64.10.9.28/32;
  address src_addr3 10.10.10.10/24;
  address bbc {
    dns-name www.bbc.com;
  }
  address-set my_source_addresses {
    address src_addr1;
    address src_addr2;
    address src_addr3;
  }
}
```

Meaning The output displays information about all the addresses configured in an address book in the specified. Verify the following information:

- Configured addresses belong to the correct address book.
- Configured address book belongs to the correct zone.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Security Policy Address Books and Address Sets Overview on page 103
- Limitations of Addresses and Address Sets on page 107
- Example: Configuring Address Books on page 108
- Example: Configuring Schedulers (CLI) on page 136

PART 3

Security Policies

- Security Policies on page 115
- Security Policy Schedulers on page 135
- Security Policy Applications on page 139

CHAPTER 6

Security Policies

- Security Policies Overview on page 115
- Understanding Security Policy Rules on page 118
- Understanding Security Policy Elements on page 120
- Security Policies Configuration Overview on page 120
- Example: Configuring a Security Policy to Permit or Deny All Traffic on page 121
- Example: Configuring a Security Policy to Permit or Deny Selected Traffic on page 125
- Understanding Security Policy Ordering on page 129
- Example: Reordering the Policies on page 130
- Troubleshooting Security Policies on page 131
- Monitoring Policy Statistics on page 132
- Matching Security Policies on page 133

Security Policies Overview

With the advent of the Internet, the need for a secure network has become vital for businesses with an Internet connection. Before a network can be secured for a business, a network security policy has to outline all the network resources within that business and identify the required security level for those resources. The network security policy also defines the security threats and the actions taken for such threats. Junos OS stateful firewall policy provides a set of tools to network administrators, enabling them to implement network security for their organizations.

In a Junos OS stateful firewall, the security policies enforce rules for transit traffic, in terms of what traffic can pass through the firewall, and the actions that need to take place on traffic as it passes through the firewall. From the perspective of security policies, the traffic enters one security zone and exits another security zone. This combination of a *from-zone* and *to-zone* is called a *context*. Each context contains an *ordered list* of policies. Each policy is processed in the order that it is defined within a context.

A security policy, which can be configured from the user interface, controls the traffic flow from one zone to another zone by defining the kind(s) of traffic permitted from specified IP sources to specified IP destinations at scheduled times.

Policies allow you to deny, permit, reject (deny and send a TCP RST or ICMP port unreachable message to the source host), encrypt and decrypt, authenticate, prioritize, schedule, filter, and monitor the traffic attempting to cross from one security zone to another. You decide which users and what data can enter and exit, and when and where they can go.



NOTE: For a J Series or an SRX Series device that supports virtual systems, policies set in the root system do not affect policies set in virtual systems.

A J Series or an SRX Series device secures a network by inspecting, and then allowing or denying, all connection attempts that require passage from one security zone to another.

Logging capability can also be enabled with security policies during session initialization (**session-init**) or session close (**session-close**) stage.

- To view logs from denied connections, enable log on **session-init**.
- To log sessions after their conclusion/tear-down, enable log on **session-close**.



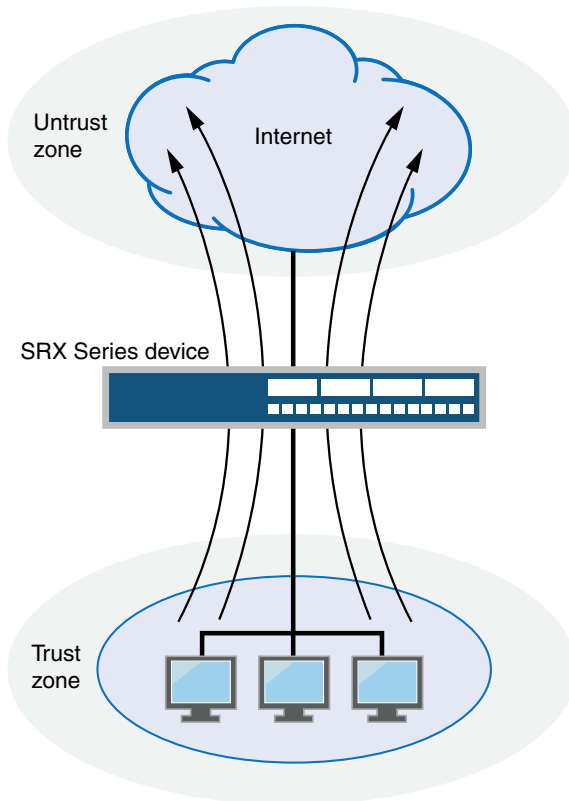
NOTE: Session log is enabled at real time in the flow code which impacts the user performance. If both **session-close** and **session-init** are enabled, performance is further degraded as compared to enabling **session-init** only.

By default, a device denies all traffic in all directions, including intra-zone and inter-zone directions. Through the creation of policies, you can control the traffic flow from zone to zone by defining the kinds of traffic permitted to pass from specified sources to specified destinations at scheduled times.

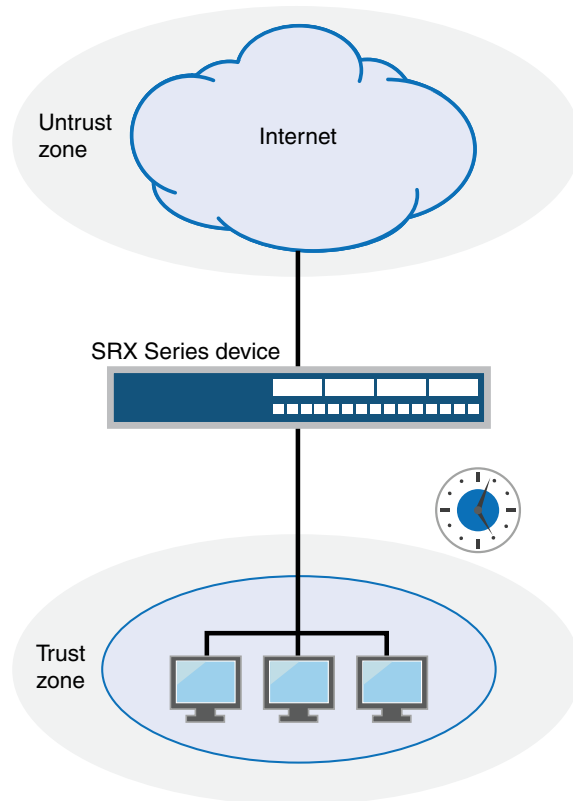
At the broadest level, you can allow all kinds of traffic from any source in one zone to any destination in all other zones without any scheduling restrictions. At the narrowest level, you can create a policy that allows only one kind of traffic between a specified host in one zone and another specified host in another zone during a scheduled interval of time. See Figure 10 on page 117.

Figure 10: Default Policy

Broadly defined Internet access: Any service from any point in the trust zone to any point in the untrust zone at any time.



Narrowly defined Internet access: SMTP service from a mail server in the trust zone to a mail server in the untrust zone from 5:00 AM to 7:00 PM.



g030677

Every time a packet attempts to pass from one zone to another or between two interfaces bound to the same zone, the device checks for a policy that permits such traffic (see “Understanding Security Zones” on page 87 and “Policy Application Sets Overview” on page 140). To allow traffic to pass from one security zone to another—for example, from zone A to zone B—you must configure a policy that permits zone A to send traffic to zone B. To allow traffic to flow the other way, you must configure another policy permitting traffic from zone B to zone A.

To allow data traffic to pass between zones, you must configure firewall policies.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Security Policy Rules on page 118
 - Understanding Security Policy Elements on page 120
 - Security Policies Configuration Overview on page 120
 - Understanding Security Policy Ordering on page 129
 - Security Zones and Interfaces Overview on page 85

Understanding Security Policy Rules

The security policy applies the security rules to the transit traffic within a context (**from-zone** to **to-zone**). Each policy is uniquely identified by its name. The traffic is classified by matching its source and destination zones, the source and destination addresses, and the application that the traffic carries in its protocol headers with the policy database in the data plane.

Each policy is associated with the following characteristics:

- A source zone
- A destination zone
- One or many source address names or address set names
- One or many destination address names or address set names
- One or many application names or application set names

These characteristics are called the *match criteria*. Each policy also has actions associated with it: permit, deny, reject, count, and log. You have to specify the match condition arguments when you configure a policy, source address, destination address, and application name.

You can specify to configure a policy with IPv4 or IPv6 addresses using the wildcard entry **any**. When flow support is not enabled for IPv6 traffic, **any** matches IPv4 addresses. For example, if you want to include both IPV4 and IPv6 addresses in the match criteria, then **any** is used. You can also specify the wildcard **any-ipv4** or **any-ipv6** for the source and destination address match criteria to include only IPv4 or only IPv6 addresses, respectively.

If you do not want to specify a specific application, enter **any** as the default application. To look up the default applications, from configuration mode, enter **show groups junos-defaults | find applications (predefined applications)**. For example, if you do not supply an application name, the policy is installed with the application as a wildcard (default). Therefore, any data traffic that matches the rest of the parameters in a given policy would match the policy regardless of the application type of the data traffic.

The action of the first policy that the traffic matches is applied to the packet. If there is no matching policy, the packet is dropped. Policies are searched from top to bottom, so it is a good idea to place more specific policies near the top of the list. You should also place IPsec VPN tunnel policies near the top. Place the more general policies, such as one that would allow certain users access to all Internet applications, at the bottom of the list. For example, place deny-all or reject-all policies at the bottom after all of the specific policies have been parsed before and legitimate traffic has been allowed/count/logged.

Policies are looked up during flow processing after firewall filters and screens have been processed and route look up has been completed by the Services Processing Unit (SPU) (for high-end SRX devices). Policy look up determines the destination zone, destination address, and egress interface.

When you are creating a policy, the following policy rules apply:

- Security policies are configured in a **from-zone** to **to-zone** direction. Under a specific zone direction, each security policy contains a name, match criteria, an action, and miscellaneous options.
- The policy name, match criteria, and action are required.
- The policy name is a keyword.
- The source address in the match criteria is composed of one or more address names or address set names in the **from-zone**.
- The destination address of the match criteria is composed of one or more address names or address set names in the **to-zone**.
- The application name in the match criteria is composed of the name of one or more applications or application sets.
- One of the following actions is required: permit, deny, reject, count, or log.
- When logging is enabled, the system logs at session close (**session-close**) time by default. To enable logging at session creation, use the **session-init** command.
- When the count alarm is turned on, you can, optionally, specify alarm thresholds in bytes per second and kilobytes per minute.
- You cannot specify **global** as either the **from-zone** or the **to-zone** except under following condition:

Any policy configured with the **to-zone** as a global zone must have a single destination address to indicate that either static NAT or incoming NAT has been configured in the policy.

- In SRX Series Services Gateways, the policy permit option with NAT is simplified. Each policy will optionally indicate whether it allows NAT translation, does not allow NAT translation, or does not care.
- Address names cannot begin with the following reserved prefixes. These are used only for address NAT configuration:
 - **static_nat_**
 - **incoming_nat_**
 - **junos_**
- Application names cannot begin with the **junos_** reserved prefix.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Understanding Security Policy Elements on page 120
 - Security Policies Configuration Overview on page 120
 - Understanding Security Policy Ordering on page 129

Understanding Security Policy Elements

A policy permits, denies, or tunnels specified types of traffic unidirectionally between two points.

Each policy consists of:

- A unique name for the policy.
- A **from-zone** and a **to-zone**, for example: `user@host# set security policy from-zone untrust to-zone untrust`
- A set of match criteria defining the conditions that must be satisfied to apply the policy rule. The match criteria are based on a source IP address, destination IP address, and applications.
- A set of actions to be performed in case of a match—permit, deny, or reject.
- Accounting and auditing elements—counting, logging, or structured system logging.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - *Junos OS CLI Reference*
 - Security Policies Overview on page 115
 - Understanding Security Policy Rules on page 118
 - Security Policies Configuration Overview on page 120
 - Understanding Security Policy Ordering on page 129

Security Policies Configuration Overview

You must complete the following tasks to create a security policy:

1. Create zones. See “Example: Creating Security Zones” on page 88
2. Configure an address book with addresses for the policy. See “Example: Configuring Address Books” on page 108
3. Create an application (or application set) that indicates that the policy applies to traffic of that type. See “Example: Configuring Applications and Application Sets” on page 141
4. Create the policy. See “Example: Configuring a Security Policy to Permit or Deny All Traffic” on page 121 and “Example: Configuring a Security Policy to Permit or Deny Selected Traffic” on page 125
5. Create schedulers if you plan to use them for your policies. See “Example: Configuring Schedulers (CLI)” on page 136
6. Bind a policy to a scheduler. See “Example: Associating a Policy to a Scheduler (CLI)” on page 137

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Security Policy Rules on page 118
 - Understanding Security Policy Elements on page 120
 - Troubleshooting Security Policies on page 131

Example: Configuring a Security Policy to Permit or Deny All Traffic

This example shows how to configure a security policy to permit or deny traffic.

- Requirements on page 121
- Overview on page 121
- Configuration on page 122
- Verification on page 124

Requirements

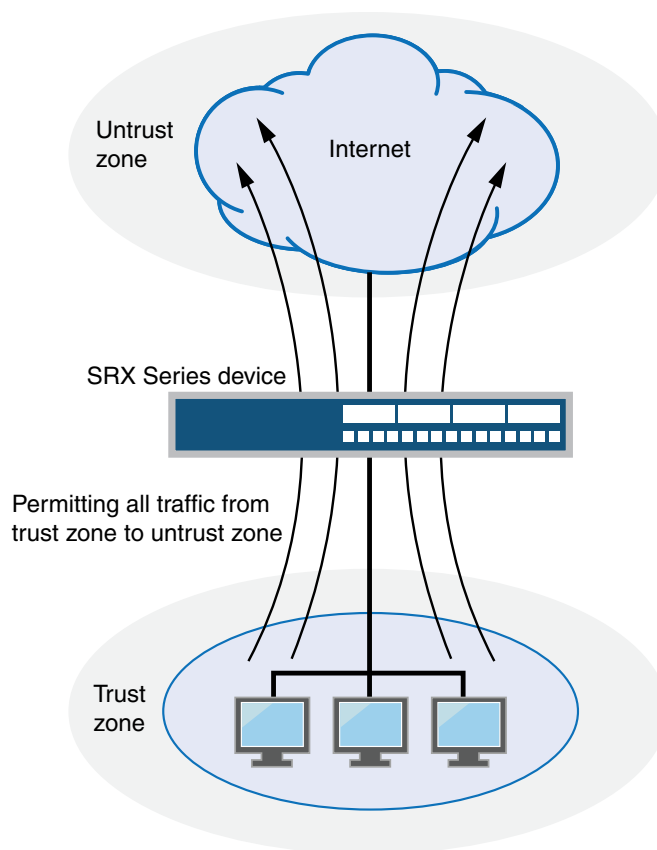
Before you begin:

- Create zones. See “Example: Creating Security Zones” on page 88.
- Configure an address book and create addresses for use in the policy. See “Example: Configuring Address Books” on page 108.
- Create an application (or application set) that indicates that the policy applies to traffic of that type. See “Example: Configuring Applications and Application Sets” on page 141.

Overview

In a Junos OS, security policies enforce rules for transit traffic, in terms of what traffic can pass through the device, and the actions that need to take place on traffic as it passes through the device. From the perspective of security policies, the traffic enters one security zone and exits another security zone. In this example, you configure the trust and untrust interfaces, ge-0/0/2 and ge-0/0/1. See Figure 11 on page 122.

Figure 11: Permitting All Traffic



This configuration example shows how to:

- Permit or deny all traffic from the trust zone to the untrust zone but block everything from the untrust zone to the trust zone.
- Permit or deny selected traffic from a host in the trust zone to a server in the untrust zone at a particular time.

Configuration

CLI Quick Configuration

To quickly configure a security policy to permit or deny all traffic, make sure the correct interfaces are used. Copy the following commands and paste them into the CLI:

```
[edit]
set security zones security-zone trust interfaces ge-0/0/2 host-inbound-traffic
  system-services all
set security zones security-zone untrust interfaces ge-0/0/1 host-inbound-traffic
  system-services all
set security policies from-zone trust to-zone untrust policy permit-all match
  source-address any
set security policies from-zone trust to-zone untrust policy permit-all match
  destination-address any
set security policies from-zone trust to-zone untrust policy permit-all match application
  any
```

```

set security policies from-zone trust to-zone untrust policy permit-all set then permit
set security policies from-zone untrust to-zone trust policy deny-all match source-address
any
set security policies from-zone untrust to-zone trust policy deny-all match
destination-address any
set security policies from-zone untrust to-zone trust policy deny-all match application
any
set security policies from-zone untrust to-zone trust policy deny-all then deny

```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a security policy to permit or deny all traffic:

1. Configure the interfaces and security zones.

```

[edit security zones]
user@host# set security-zone trust interfaces ge-0/0/2 host-inbound-traffic
system-services all
user@host# set security-zone untrust interfaces ge-0/0/1 host-inbound-traffic
system-services all

```

2. Create the security policy to permit traffic from the trust zone to the untrust zone.

```

[edit security policies from-zone trust to-zone untrust]
user@host# set policy permit-all match source-address any
user@host# set policy permit-all match destination-address any
user@host# set policy permit-all match application any
user@host# set policy permit-all then permit

```

3. Create the security policy to deny traffic from the untrust zone to the trust zone.

```

[edit security policies from-zone untrust to-zone trust]
user@host# set policy deny-all match source-address any
user@host# set policy deny-all match destination-address any
user@host# set policy deny-all match application any
user@host# set policy deny-all then deny

```

Results From configuration mode, confirm your configuration by entering the **show security policies** and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.



NOTE: The configuration example is a default permit-all from the trust zone to the untrust zone.

```

[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy permit-all {
    match {
      source-address any;

```

```
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
}
from-zone untrust to-zone trust {
    policy deny-all {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            deny;
        }
    }
}

user@host# show security zones
security-zone trust {
    interfaces {
        ge-0/0/2.0 {
            host-inbound-traffic {
                system-services {
                    all;
                }
            }
        }
    }
}
security-zone untrust {
    interfaces {
        ge-0/0/1.0 {
            host-inbound-traffic {
                system-services {
                    all;
                }
            }
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Policy Configuration on page 124

Verifying Policy Configuration

Purpose Verify information about address books and zones.

Action From operational mode, enter the **show security policies policy-name permit-all detail** command to display a summary of all security policies configured on the device.

Meaning The output displays information about policies configured on the system. Verify the following information:

- From and to zones
- Source and destination addresses
- Match criteria

Example: Configuring a Security Policy to Permit or Deny Selected Traffic

This example shows how to configure a security policy to permit or deny selected traffic.

- Requirements on page 125
- Overview on page 125
- Configuration on page 126
- Verification on page 128

Requirements

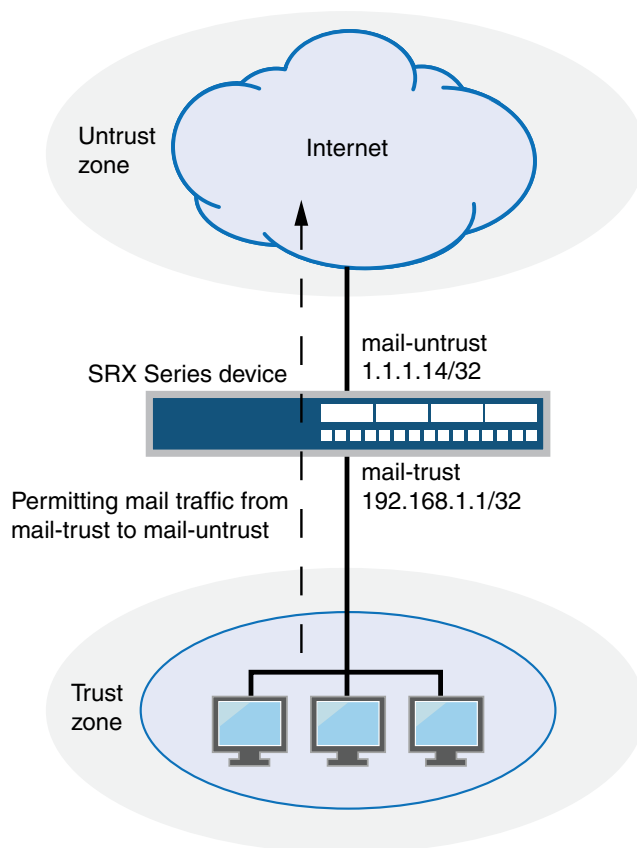
Before you begin:

- Create zones. See “Example: Creating Security Zones” on page 88.
- Configure an address book and create addresses for use in the policy. See “Example: Configuring Address Books” on page 108.
- Create an application (or application set) that indicates that the policy applies to traffic of that type. See “Example: Configuring Applications and Application Sets” on page 141.
- Permit traffic to and from trust and untrust zones. See “Example: Configuring a Security Policy to Permit or Deny All Traffic” on page 121.

Overview

In a Junos OS, security policies enforce rules for the transit traffic, in terms of what traffic can pass through the device, and the actions that need to take place on the traffic as it passes through the device. From the perspective of security policies, the traffic enters one security zone and exits another security zone. In this example, you configure a specific security to allow only e-mail traffic from a host in the trust zone to a server in the untrust zone. No other traffic is allowed. See Figure 12 on page 126.

Figure 12: Permitting Selected Traffic



Configuration

CLI Quick Configuration

To quickly configure a security policy to allow selected traffic, copy the following commands and paste them into the CLI:

```
[edit]
set security zones security-zone trust interfaces ge-0/0/2 host-inbound-traffic
  system-services all
set security zones security-zone untrust interfaces ge-0/0/1 host-inbound-traffic
  system-services all
set security zones security-zone untrust address-book address mail-untrust 1.1.1.24/32
set security zones security-zone trust address-book address mail-trust 192.168.1.1/32
set security policies from-zone trust to-zone untrust policy permit-mail match
  source-address mail-trust
set security policies from-zone trust to-zone untrust policy permit-mail match
  destination-address mail-untrust
set security policies from-zone trust to-zone untrust policy permit-mail match application
  junos-mail
set security policies from-zone trust to-zone untrust policy permit-mail then permit
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a security policy to allow selected traffic:

1. Configure the interfaces and security zones.

```
[edit security zones]
user@host# set security-zone trust interfaces ge-0/0/2 host-inbound-traffic
system-services all
user@host# set security-zone untrust interfaces ge-0/0/1 host-inbound-traffic
system-services all
```

2. Create address book entries for both client and server.

```
[edit security zones]
user@host# set security-zone untrust address-book address mail-untrust 1.1.1.24/32
user@host# set security-zone trust address-book address mail-trust 192.168.1.1/32
```

3. Define the policy to permit mail traffic.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy permit-mail match source-address mail-trust
user@host# set policy permit-mail match destination-address mail-untrust
user@host# set policy permit-mail match application junos-mail
user@host# set policy permit-mail then permit
```

Results From configuration mode, confirm your configuration by entering the **show security policies** and **show security zones** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy permit-mail {
    match {
      source-address mail-trust;
      destination-address mail-untrust;
      application junos-mail;
    }
    then {
      permit;
    }
  }
}

user@host# show security zones
security-zone trust {
  address-book {
    address mail-trust 192.168.1.1/32;
  }
  host-inbound-traffic {
    system-services {
      all;
    }
  }
}
```

```
interfaces {
  ge-0/0/2 {
    host-inbound-traffic {
      system-services {
        all;
      }
    }
  }
}
security-zone untrust {
  address-book {
    address mail-untrust 1.1.1.24/32;
  }
  interfaces {
    ge-0/0/1 {
      host-inbound-traffic {
        system-services {
          all;
        }
      }
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Policy Configuration on page 128

Verifying Policy Configuration

Purpose Verify information about address books and zones.

Action From operational mode, enter the **show security policies policy-name permit-all detail** command to display a summary of all security policies configured on the device.

Meaning The output displays information about policies configured on the system. Verify the following information:

- From and to zones
- Source and destination addresses
- Match criteria

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Security Policies Overview on page 115
- Example: Configuring a Security Policy to Permit or Deny All Traffic on page 121

Understanding Security Policy Ordering

Junos OS offers a tool for verifying that the order of policies in the policy list is valid.

It is possible for one policy to eclipse, or *shadow*, another policy. Consider the following examples:

Example 1

```
[edit]
user@host# set security zones security-zone trust interfaces ge-0/0/20
    host-inbound-traffic system-services all
user@host# set security zones security-zone untrust interfaces ge-0/0/1
    host-inbound-traffic system-services all
user@host# set security policies from-zone trust to-zone untrust policy permit-all match
    source-address any
user@host# set security policies from-zone trust to-zone untrust match
    destination-address any
user@host# set security policies from-zone trust to-zone untrust match application any
user@host# set security policies from-zone trust to-zone untrust set then permit
user@host# set security policies from-zone untrust to-zone trust policy deny-all match
    source-address any
user@host# set security policies from-zone untrust to-zone trust policy deny-all match
    destination-address any
user@host# set security policies from-zone untrust to-zone trust policy deny-all match
    application any
user@host# set security policies from-zone untrust to-zone trust policy deny-all then
    deny
```

Example 2

```
[edit]
user@host# set security zones security-zone trust interfaces ge-0/0/2.0
    host-inbound-traffic system-services all
user@host# set security zones security-zone untrust interfaces ge-0/0/1.0
    host-inbound-traffic system-services all
user@host# set security zones security-zone untrust address-book address mail-untrust
    1.1.1.24/32
user@host# set security zones security-zone trust address-book address mail-trust
    192.168.1.1/32
user@host# set security policies from-zone trust to-zone untrust policy permit-mail match
    source-address mail-trust
user@host# set security policies from-zone trust to-zone untrust policy permit-mail match
    destination-address mail-untrust
user@host# set security policies from-zone trust to-zone untrust policy permit-mail match
    application junos-mail
user@host# set security policies from-zone trust to-zone untrust policy permit-mail then
    permit
```

In examples 1 and 2, where policy **permit-mail** is configured after policy **permit-all** from zone **trust** to zone **untrust**. All traffic coming from zone **untrust** matches the first policy **permit-all** and is allowed by default. No traffic matches policy **permit-mail**.

Because Junos OS performs a policy lookup starting from the top of the list, when it finds a match for traffic received, it does not look any lower in the policy list. To correct the

pervious example, you can simply reverse the order of the policies, putting the more specific one first:

```
[edit]
user@host# insert security policies from-zone trust to-zone untrust policy permit-mail
before policy permit-all
```

In cases where there are dozens or hundreds of policies, the eclipsing of one policy by another might not be so easy to detect. To check if policies are being shadowed, enter the following command:

```
[edit]
user@host# show policy-options <policy-name>
```

This command reports the shadowing and shadowed policies. It is then the administrator's responsibility to correct the situation.



NOTE: The concept of *policy shadowing* refers to the situation where a policy higher in the policy list always takes effect before a subsequent policy. Because the policy lookup always uses the first policy it finds that matches the five-part tuple of the source and destination zone, source and destination address, and application type, if another policy applies to the same tuple (or a subset of the tuple), the policy lookup uses the first policy in the list and never reaches the second one.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Configuration Overview on page 120
 - Example: Configuring a Security Policy to Permit or Deny All Traffic on page 121
 - Example: Configuring a Security Policy to Permit or Deny Selected Traffic on page 125

Example: Reordering the Policies

This example shows how to move policies around after they have been created.

- Requirements on page 130
- Overview on page 131
- Configuration on page 131
- Verification on page 131

Requirements

Before you begin:

- Create zones. See “Example: Creating Security Zones” on page 88.
- Configure the address book and create addresses for use in the policy. See “Example: Configuring Address Books” on page 108.

Overview

To reorder policies to correct shadowing, you can simply reverse the order of the policies, putting the more specific one first.

Configuration

Step-by-Step Procedure

To reorder existing policies:

1. Reorder two existing policies by entering the following command:

```
[edit]  
user@host# insert security policies from-zone trust to-zone untrust policy  
permit-mail before policy permit-all
```
2. If you are done configuring the device, commit the configuration.

```
[edit]  
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security policies** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Understanding Security Policy Ordering on page 129

Troubleshooting Security Policies

- Checking a Security Policy Commit Failure on page 131
- Verifying a Security Policy Commit on page 132
- Debugging Policy Lookup on page 132

Checking a Security Policy Commit Failure

Problem Most policy configuration failures occur during a commit or runtime.

Commit failures are reported directly on the CLI when you execute the CLI command **commit-check** in configuration mode. These errors are configuration errors, and you cannot commit the configuration without fixing these errors.

Solution To fix these errors, do the following:

1. Review your configuration data.
2. Open the file `/var/log/nsd_chk_only`. This file is overwritten each time you perform a commit check and contains detailed failure information.

Verifying a Security Policy Commit

Problem Upon performing a policy configuration commit, if you notice that the system behavior is incorrect, use the following steps to troubleshoot this problem:

Solution

1. Operational **show** Commands—Execute the operational commands for security policies and verify that the information shown in the output is consistent with what you expected. If not, the configuration needs to be changed appropriately.
2. Traceoptions—Set the **traceoptions** command in your policy configuration. The flags under this hierarchy can be selected as per user analysis of the **show** command output. If you cannot determine what flag to use, the flag option **all** can be used to capture all trace logs.

```
user@host# set security policies traceoptions <flag all>
```

You can also configure an optional filename to capture the logs.

```
user@host# set security policies traceoptions <filename>
```

If you specified a filename in the trace options, you can look in the `/var/log/<filename>` for the log file to ascertain if any errors were reported in the file. (If you did not specify a filename, the default filename is `eventd`.) The error messages indicate the place of failure and the appropriate reason.

After configuring the trace options, you must recommit the configuration change that caused the incorrect system behavior.

Debugging Policy Lookup

Problem When you have the correct configuration, but some traffic was incorrectly dropped or permitted, you can enable the **lookup** flag in the security policies traceoptions. The **lookup** flag logs the lookup related traces in the trace file.

Solution

```
user@host# set security policies traceoptions <flag lookup>
```

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Security Policies Overview on page 115
- Checking a Security Policy Commit Failure on page 131
- Verifying a Security Policy Commit on page 132
- Debugging Policy Lookup on page 132
- Monitoring Policy Statistics on page 132

Monitoring Policy Statistics

Purpose Monitor and record traffic that Junos OS permits or denies based on previously configured policies.

Action To monitor traffic, enable the count and log options.

Count—Configurable in an individual policy. If count is enabled, counters are collected for the number of packets, bytes, and sessions that enter the firewall for a given policy. For counts (only for packets and bytes), you can specify that alarms be generated whenever the traffic exceeds specified thresholds.

Log—Logging capability can be enabled with security policies during session initialization (**session-init**) or session close (**session-close**) stage.

- To view logs from denied connections, enable log on **session-init**.
- To log sessions after their conclusion/tear-down, enable log on **session-close**.



NOTE: Session log is enabled at real time in the flow code which impacts the user performance. If both **session-close** and **session-init** are enabled, performance is further degraded as compared to enabling **session-init** only.

For details about information collected for session logs, see “Information Provided in Session Log Entries for SRX Series Services Gateways” on page 15.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Troubleshooting Security Policies on page 131
 - Checking a Security Policy Commit Failure on page 131
 - Verifying a Security Policy Commit on page 132
 - Debugging Policy Lookup on page 132

Matching Security Policies

The **show security match-policies** command allows you to troubleshoot traffic problems in the five tuples: source port, destination port, source IP address, destination IP address, and protocol. For example, if your traffic is not passing because either a correct policy is not configured or the source of the traffic is incorrect, then the **show security match-policies** command allows you to work offline and identify where the problem actually exists. It uses the search engine to identify the problem and thus enables you to use the appropriate match policy for the traffic.



NOTE: The **show security match-policies** command is applicable only to security policies; IDP policies are not supported. Only the first matched policy is returned.

**show security
match-policies**

```
user@host> show security match-policies
From-zone: z1, To-zone: z2
source-ip 10.10.10.1 destination-ip 30.30.30.1 source-port 1 destination-port
```

```
21 protocol tcp
Policy: p1, action-type: permit, State: enabled, Index: 4, AI: disabled, Scope
Policy 0
Policy Type: Configured
Sequence number: 1
From zone: z1, To zone: z2
Source addresses:
  a2: 20.20.0.0/16
  a3: 10.10.10.1/32
Destination addresses:
  d2: 40.40.0.0/16
  d3: 30.30.30.1/32
Application: junos-ftp
  IP protocol: tcp, ALG: ftp, Inactivity timeout: 1800
  Source port range: [0-0]
  Destination port range: [21-21]
Intrusion Detection and Prevention: enabled
Unified Access Control: enabled
```

For more information on matching policies and a description of the output fields, see the *Junos OS CLI Reference*.

CHAPTER 7

Security Policy Schedulers

- Security Policy Schedulers Overview on page 135
- Example: Configuring Schedulers (CLI) on page 136
- Example: Associating a Policy to a Scheduler (CLI) on page 137
- Verifying Scheduled Policies on page 137

Security Policy Schedulers Overview

Schedulers are powerful features that allow a policy to be activated for a specified duration. You can define schedulers for a single (nonrecurrent) or recurrent time slot within which a policy is active. You can create schedulers irrespective of a policy, meaning that a scheduler cannot be used by any policies. However, if you want a policy to be active within a scheduled time, then you must first create a scheduler.

When a scheduler times out, the associated policy is deactivated and all sessions associated with the policy are also timed out.

If a policy contains a reference to a scheduler, the schedule determines when the policy is active, that is, when it can be used as a possible match for traffic. Schedulers allow you to restrict access to a resource for a period of time or remove a restriction.

The following guidelines apply to schedulers:

- A scheduler can have multiple policies associated with it; however, a policy cannot be associated with multiple schedulers.
- A policy is active during the time when the scheduler it refers to is also active.
- When a scheduler is off, the policy is unavailable for policy lookup.
- A scheduler can be configured as one of the following:
 - Scheduler can be active for a single time slot, as specified by a start date and time and a stop date and time.
 - Scheduler can be active forever (recurrent), but as specified by the daily schedule. The schedule on a specific day (time slot) takes priority over the daily schedule.
 - Scheduler can be active within a time slot as specified by the weekday schedule.
 - Scheduler can have a combination of two time slots (daily and timeslot).

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Example: Configuring Schedulers (CLI) on page 136
 - Example: Associating a Policy to a Scheduler (CLI) on page 137
 - Verifying Scheduled Policies on page 137

Example: Configuring Schedulers (CLI)

To configure a scheduler, you enter a meaningful name and a start and stop time for the scheduler. You can also attach comments.

To configure a scheduler using the CLI configuration editor:

1. Use the following commands to set a schedule that allows a policy, which refers to it, to be used for packet match checks from 8 AM to 9 PM all days of the week from October 1, 2007 to April 1, 2008 except Saturdays and Sundays. Otherwise, the policy is inactive.

```
user@host# set schedulers scheduler sch1 start-date 2007-10-01.08:00 stop-date 2008-04-01.21:00
user@host# set schedulers scheduler sch1 saturday exclude
user@host# set schedulers scheduler sch1 sunday exclude
```

2. Use the following command to associate the schedule to the policy, allowing access during regular work hours, as specified.

```
user@host# set security policies from-zone green to-zone red policy abc
scheduler-name sch1
```

3. Use the following commands to set schedulers that allow associated policies to check for packet matches from noon to 6 PM on Saturdays and Sundays. Otherwise the policy is inactive.

```
user@host# set schedulers scheduler SatHrs saturday start-time 12:00:00 stop-time 18:00:00
user@host# set schedulers scheduler SunHrs sunday start-time 12:00 stop-time 18:00
```

4. Use the following commands to bind these schedules to the policy, allowing access during the specified weekend hours.

```
user@host# set security policies from-zone green to-zone red policy abc
scheduler-name SatHrs
user@host# set security policies from-zone green to-zone red policy abc
scheduler-name SunHrs
```

5. If you are finished configuring the device, commit the configuration.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Example: Associating a Policy to a Scheduler (CLI) on page 137

- Verifying Scheduled Policies on page 137

Example: Associating a Policy to a Scheduler (CLI)

A scheduler is referred by security policies to activate or deactivate a policy according to scheduled times. You can associate a policy with a scheduler as you create the policy.

In the following example, you configure schedulers that allow associated policies to be used to check for packet matches from noon to 6 PM on Saturdays and Sundays. Otherwise the policy is inactive.

```
user@host# set schedulers scheduler SatHrs saturday start-time 12:00 stop-time 18:00
user@host# set schedulers scheduler SunHrs sunday start-time 12:00 stop-time 18:00
```

The following commands bind these schedules to policy **abc** allowing access during the specified weekend hours.

```
user@host# set security policies from-zone green to-zone red policy abc scheduler-name
SatHrs
user@host# set security policies from-zone green to-zone red policy abc scheduler-name
SunHrs
```

If you are finished configuring the device, commit the configuration.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Example: Configuring Schedulers (CLI) on page 136
 - Verifying Scheduled Policies on page 137

Verifying Scheduled Policies

Purpose Display information about address books and zones.

Action Use the **show schedulers** CLI command to display information about schedulers configured on the system. If a specific scheduler is identified, detailed information is displayed for that scheduler only.

```
user@host# show schedulers
scheduler sche1 {
    /* This is sched1 */
    start-date 2006-11-02.12:12 stop-date 2007-11-02.12:11;
}
scheduler sche2 {
    daily {
        all-day;
    }
    sunday {
        start-time 16:00 stop-time 17:00;
    }
    friday {
        exclude;
    }
}
```

```
scheduler sche3 {  
  start-date 2006-11-02.12:12 stop-date 2007-11-02.12:11;  
  daily {  
    start-time 10:00 stop-time 17:00  
  }  
  sunday {  
    start-time 12:00 stop-time 14:00;  
    start-time 16:00 stop-time 17:00;  
  }  
  monday {  
    all-day;  
  }  
  friday {  
    exclude;  
  }  
}
```

Meaning The output displays information about schedulers configured on the system. Verify the following information:

- Daily (recurrent) and one-time only (nonrecurrent) schedulers are configured correctly.
- Schedulers are active if policies are associated.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Example: Configuring Schedulers (CLI) on page 136
 - Example: Associating a Policy to a Scheduler (CLI) on page 137

CHAPTER 8

Security Policy Applications

- Security Policy Applications Overview on page 139
- Policy Application Sets Overview on page 140
- Example: Configuring Applications and Application Sets on page 141
- Custom Policy Applications on page 142
- Policy Application Timeouts on page 146
- Understanding the ICMP Predefined Policy Application on page 150
- Default Behaviour of ICMP Unreachable Errors on page 154
- Understanding Internet-Related Predefined Policy Applications on page 155
- Understanding Microsoft Predefined Policy Applications on page 156
- Understanding Dynamic Routing Protocols Predefined Policy Applications on page 157
- Understanding Streaming Video Predefined Policy Applications on page 158
- Understanding Sun RPC Predefined Policy Applications on page 159
- Understanding Security and Tunnel Predefined Policy Applications on page 160
- Understanding IP-Related Predefined Policy Applications on page 160
- Understanding Instant Messaging Predefined Policy Applications on page 161
- Understanding Management Predefined Policy Applications on page 162
- Understanding Mail Predefined Policy Applications on page 163
- Understanding UNIX Predefined Policy Applications on page 164
- Understanding Miscellaneous Predefined Policy Applications on page 164

Security Policy Applications Overview

Applications are types of traffic for which protocol standards exist. Each application has a transport protocol and destination port number(s) associated with it, such as TCP/port 21 for FTP and TCP/port 23 for Telnet. When you create a policy, you must specify an application for it.

You can select one of the predefined applications from the application book, or a custom application or application set that you created. You can see which application you can use in a policy by using the **show application** CLI command.



NOTE: Each predefined application has a source port range of 1–65535, which includes the entire set of valid port numbers. This prevents potential attackers from gaining access by using a source port outside of the range. If you need to use a different source port range for any predefined application, create a custom application. For information, see “Understanding Custom Policy Applications” on page 142.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Understanding Security Policy Rules on page 118
 - Understanding Security Policy Elements on page 120
 - Policy Application Sets Overview on page 140

Policy Application Sets Overview

When you create a policy, you must specify an application, or service, for it to indicate that the policy applies to traffic of that type. Sometimes the same applications or a subset of them can be present in multiple policies, making it difficult to manage. Junos OS allows you to create groups of applications called *application sets*. Application sets simplify the process by allowing you to manage a small number of application sets, rather than a large number of individual application entries.

The application (or application set) is referred to by security policies as match criteria for packets initiating sessions. If the packet matches the application type specified by the policy and all other criteria match, then the policy action is applied to the packet.

You can specify the name of an application set in a policy. In this case, if all of the other criteria match, any one of the applications in the application set serves as valid matching criteria; **any** is the default application name that indicates all possible applications.

Applications are created in the `.../applications/application/application-name` directory. You do not need to configure an application for any of the services that are predefined by the system.

In addition to predefined services, you can configure a custom service. After you create a custom service, you can refer to it in a policy.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Custom Application Mappings on page 142
 - Understanding Policy Application Timeout Configuration and Lookup on page 146
 - Example: Configuring Applications and Application Sets on page 141

Example: Configuring Applications and Application Sets

This example shows how to configure applications and application sets.

- Requirements on page 141
- Overview on page 141
- Configuration on page 141
- Verification on page 142

Requirements

Before you begin, configure the required applications. See “Policy Application Sets Overview” on page 140.

Overview

Rather than creating or adding multiple individual application names to a policy, you can create an application set and refer to the name of the set in a policy. For example, for a group of employees, you can create an application set that contains all the approved applications.

In this example, you create an application set that are used to log into the servers in the ABC (intranet) zone, to access the database, and to transfer files.

- Define the applications in the configured application set.
- Managers in zone A and managers in zone B use these services. Therefore, give the application set a generic name, such as MgrAppSet.
- Create an application set for the applications that are used for e-mail and Web-based applications that are delivered by the two servers in the external zone.

Configuration

Step-by-Step Procedure

To configure an application and application set:

1. Create an application set for managers.

```
[edit applications]
user@host# set application-set MgrAppSet application junos-ssh
user@host# set application-set MgrAppSet application junos-telnet
```
2. Create another application set for e-mail and Web-based Example: Setting a Policy Application Time applications.

```
[edit applications]
user@host# set application-set WebMailApps application junos-smtp
user@host# set application-set WebMailApps application junos-pop3
```
3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show applications** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Security Policy Applications Overview on page 139

Custom Policy Applications

- Understanding Custom Policy Applications on page 142
- Custom Application Mappings on page 142
- Example: Adding and Modifying Custom Policy Applications on page 143
- Example: Defining a Custom ICMP Application on page 144

Understanding Custom Policy Applications

If you do not want to use predefined applications in your policy, you can easily create custom applications.

You can assign each custom application the following attributes:

- Name
- Transport protocol
- Source and destination port numbers for applications using TCP or UDP
- Type and code values for applications using ICMP
- Timeout value

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Custom Application Mappings on page 142
 - Understanding Policy Application Timeout Configuration and Lookup on page 146
 - Understanding Policy Application Timeouts Contingencies on page 148
 - Example: Adding and Modifying Custom Policy Applications on page 143

Custom Application Mappings

The application option specifies the Layer 7 application that maps to the Layer 4 application that you reference in a policy. A predefined application already has a mapping to a Layer 7 application. However, for custom applications, you must link the application to an application explicitly, especially if you want the policy to apply an Application Layer Gateway (ALG) or deep inspection to the custom application.



NOTE: Junos OS supports ALGs for numerous applications, including DNS, FTP, H.323, HTTP, RSH, SIP, Telnet, and TFTP.

Applying an ALG to a custom application involves the following two steps:

- Define a custom application with a name, timeout value, transport protocol, and source and destination ports.
- When configuring a policy, reference that application and the application type for the ALG that you want to apply.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Security Policy Applications Overview on page 139
- Understanding Custom Policy Applications on page 142
- Understanding Policy Application Timeout Configuration and Lookup on page 146
- Understanding Policy Application Timeouts Contingencies on page 148
- Example: Adding and Modifying Custom Policy Applications on page 143

Example: Adding and Modifying Custom Policy Applications

This example shows how to add and modify custom policy applications.

- Requirements on page 143
- Overview on page 143
- Configuration on page 144
- Verification on page 144

Requirements

Before you begin, create addresses and security zones. See “Example: Creating Security Zones” on page 88.

Overview

In this example, you create a custom application using the following information:

- A name for the application, such as **cust-telnet**.
- A range of source port numbers: 1 through **65535**.
- A range of destination port numbers to receive the application request, such as 1 through **65535**.
- Whether the application uses TCP or UDP, or some other protocol as defined by the Internet specifications.

Configuration

Step-by-Step Procedure The following example requires you to navigate through various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To add and modify a custom policy application:

1. Configure TCP and specify the source port and destination port.

```
[edit applications application cust-telnet]  
user@host# set protocol tcp source-port 1-65535 destination-port 23000
```
2. Specify the length of time that the application is inactive.

```
[edit applications application cust-telnet]  
user@host# set inactivity-timeout 30
```
3. Modify a custom policy application.

```
[edit applications application cust-telnet]  
user@host# delete protocol tcp  
user@host# set application-protocol ftp
```
4. If you are done configuring the device, commit the configuration.

```
[edit]  
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show applications application** command.



NOTE: The timeout value is in minutes. If you do not set it, the timeout value of a custom application is 180 minutes. If you do not want an application to time out, type *never*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Security Policy Applications Overview on page 139
 - Understanding Custom Policy Applications on page 142
 - Example: Defining a Custom ICMP Application on page 144

Example: Defining a Custom ICMP Application

This example shows how to define a custom ICMP application.

- Requirements on page 145
- Overview on page 145

- Configuration on page 146
- Verification on page 146

Requirements

Before you begin:

- Understand custom policy application. See “Understanding Custom Policy Applications” on page 142.
- Understand the ICMP predefined policy application. See “Understanding the ICMP Predefined Policy Application” on page 150.

Overview

Junos OS supports ICMP—as well as several ICMP messages—as predefined or custom applications. When configuring a custom ICMP application, you define a type and code.

- There are different message types within ICMP. For example:
 - type 0 = Echo Request message
 - type 3 = Destination Unreachable message
- An ICMP message type can also have a message code. The code provides more specific information about the message, as shown in table Message Descriptions.

Table 14: Message Descriptions

Message Type	Message Code
5 = Redirect	0 = Redirect datagram for the network (or subnet)
	1 = Redirect datagram for the host
	2 = Redirect datagram for the type of application and network
	3 = Redirect datagram for the type of application and host
11 = Time Exceeded Codes	0 = Time to live exceeded in transit
	1 = Fragment reassembly time exceeded

Junos OS supports any type or code within the range of 0 through 55.

In this example, you define a custom application named host-unreachable using ICMP as the transport protocol. The type is 3 (for destination unreachable) and the code is 1 (for host unreachable). You set the timeout value at 4 minutes.



NOTE: For more information about ICMP types and codes, refer to RFC 792, *Internet Control Message Protocol*.

Configuration

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To define a custom ICMP application:

1. Set the application type and code.

```
[edit applications application host-unreachable]  
user@host# set icmp-type 5 icmp-code 0
```
2. Set the inactivity timeout value.

```
[edit applications application host-unreachable]  
user@host# set inactivity-timeout 4
```
3. If you are done configuring the device, commit the configuration.

```
[edit]  
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show applications** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115

Policy Application Timeouts

- Understanding Policy Application Timeout Configuration and Lookup on page 146
- Understanding Policy Application Timeouts Contingencies on page 148
- Example: Setting a Policy Application Timeout on page 149

Understanding Policy Application Timeout Configuration and Lookup

The application timeout value you set for an application determines the session timeout. You can set the timeout threshold for a predefined or custom application; you can use the application default timeout, specify a custom timeout, or use no timeout at all. Application timeout behavior is the same in virtual systems (vsys) security domains as at the root level.

Application timeout values are stored in the application entry database and in the corresponding vsys TCP and UDP port-based timeout tables. When you set an application timeout value, Junos OS updates these tables with the new value. There are also default timeout values in the applications entry database, which are taken from predefined applications. You can set a timeout, but you cannot alter a default value.

Applications with multiple rule entries share the same timeout value. If multiple applications share the same protocol and destination port range, all applications share the last timeout value configured.

For single application entries, an application timeout lookup proceeds as follows:

1. The specified timeout in the application entry database, if set.
2. The default timeout in the application entry database, if specified in the predefined application.
3. The protocol-based default timeout table. See Table 15 on page 147.

Table 15: Protocol-Based Default Timeout

Protocol	Default Timeout (seconds)
TCP	1800
UDP	60
ICMP	60
OSPF	60
Other	1800

For application groups, including hidden groups created in multicell policy configurations, and for the predefined application **ANY** (if timeout is not set), application timeout lookup proceeds as follows:

1. The vsys TCP and UDP port-based timeout table, if a timeout is set.
2. The protocol-based default timeout table.

- Related Topics
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Understanding Custom Policy Applications on page 142
 - Understanding Policy Application Timeouts Contingencies on page 148
 - Custom Application Mappings on page 142
 - Example: Adding and Modifying Custom Policy Applications on page 143

Understanding Policy Application Timeouts Contingencies

When setting timeouts, be aware of the following contingencies:

- If an application contains several application rule entries, all rule entries share the same timeout. The timeout table is updated for each rule entry that matches the protocol (for UDP and TCP—other protocols use the default). You need to define the application timeout only once. For example, if you create an application with two rules, the following commands will set the timeout to **20** seconds for both rules:

```
user@host# set applications application test protocol tcp destination-port 1035-1035  
inactivity-timeout 20
```

```
user@host# set applications application test term test protocol udp
```

```
user@host# set applications application test term test source-port 1-65535
```

```
user@host# set applications application test term test destination-port 1111-1111
```

- If multiple applications are configured with the same protocol and overlapping destination ports, the latest application timeout configured overrides the others in the port-based table. For example:

```
user@host# set applications application ftp-1 protocol tcp source-port 0-65535  
destination-port 2121-2121 inactivity-timeout 10
```

```
user@host# set applications application telnet-1 protocol tcp source-port 0-65535  
designating-port 2100-2148 inactivity-timeout 20
```

With this configuration, Junos OS applies the 20-second timeout for destination port **2121** in an application group, because the destination port numbers for telnet-1 (**2100-2148**) overlap those for ftp-1 (**2121**), and you defined telnet-1 after you defined ftp-1.

To modify an application timeout when multiple applications use the same protocol and an overlapping destination port range, you must unset the application and reset it with the new timeout value. This is because, during reboot, applications are loaded according to creation time, not modification time.

To avoid the unintended application of the wrong timeout to an application, do not create applications with overlapping destination port numbers.

- If you unset an application timeout, the default protocol-based timeout in the application entry database is used, and the timeout values in both the application entry and port-based timeout tables are updated with the default value.

If the modified application has overlapping destination ports with other applications, the default protocol-based timeout might not be the desired value. In that case, reboot Junos OS, or set the application timeout again for the desired timeout to take effect.

- When you modify a predefined application and reboot, the modified application might not be the last one in the configuration. This is because predefined applications are loaded before custom applications, and any change made to a custom application, even if made earlier, will show as later than the predefined application change when you reboot.

For example, suppose you create the following application:

```
user@host# set applications application my-application protocol tcp destination-port
179-179 inactivity-timeout 20
```

Later you modify the timeout of the predefined application BGP as follows:

```
user@host# set applications application bgp inactivity-timeout 75
```

The BGP application will use the 75-second timeout value, because it is now written to the application entry database. But the timeout for port 179, the port BGP uses, is also changed to 75 in the TCP port-based timeout table. After you reboot, the BGP application will continue to use the 75-second timeout that, as a single application, it gets from the application entry database. But the timeout in the TCP port-based table for port 179 will now be **60**. You can verify this by entering the **show applications application bgp** command.

The BGP application has no effect on single applications. But if you add BGP or my_application to an application group, the 60-second timeout value will be used for destination port 179. This is because application group timeout is taken from the port-based timeout table, if one is set.

To ensure predictability when you modify a predefined application timeout, therefore, you can create a similar application, for example:

```
user@host# set applications application my-bgp protocol tcp destination-port 179-179
inactivity-timeout 75
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Understanding Custom Policy Applications on page 142
 - Custom Application Mappings on page 142
 - Understanding Policy Application Timeout Configuration and Lookup on page 146
 - Example: Adding and Modifying Custom Policy Applications on page 143

Example: Setting a Policy Application Timeout

This example shows how to set a policy application timeout value.

- Requirements on page 149
- Overview on page 150
- Configuration on page 150
- Verification on page 150

Requirements

Before you begin, understand policy application timeouts. See “Understanding Policy Application Timeout Configuration and Lookup” on page 146.

Overview

Application timeout values are stored in the application entry database and in the corresponding vsys TCP and UDP port-based timeout tables. In this example, you set the device for a policy application timeout to 75 minutes for the FTP predefined application.

When you set an application timeout value, Junos OS updates these tables with the new value.

Configuration

Step-by-Step Procedure

To set a policy application timeout:

1. Set the inactivity timeout value.

```
[edit applications application ftp]
user@host# set inactivity-timeout 75
```

2. Commit the configuration if you are done configuring the device.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show applications** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139

Understanding the ICMP Predefined Policy Application

When you create a policy, you can specify the ICMP predefined application for the policy.

Internet Control Message Protocol (ICMP) is a part of IP and provides a way to query a network (ICMP query messages) and to receive feedback from the network for error patterns (ICMP error messages). ICMP does not, however, guarantee error message delivery or report all lost datagrams; and it is not a reliable protocol. ICMP codes and type codes describe ICMP query messages and ICMP error messages.

You can choose to permit or deny any or specific types of ICMP messages to improve network security. Some types of ICMP messages can be exploited to gain information about your network that might compromise security. For example, ICMP, TCP, or UDP packets can be constructed to return ICMP error messages that contain information about a network, such as its topology, and access list filtering characteristics. Table 16 on page 151 lists ICMP message names, the corresponding code, type, and description.

Table 16: ICMP Messages

ICMP Message Name	Type	Code	Description
ICMP-ANY	all	all	<p>ICMP-ANY affects any protocol using ICMP.</p> <p>Denying ICMP-ANY impairs any attempt to ping or monitor a network using ICMP.</p> <p>Permitting ICMP-ANY allows all ICMP messages.</p>
ICMP-ADDRESS-MASK <ul style="list-style-type: none"> Request Reply 	17 18	0 0	<p>ICMP address mask query is used for systems that need the local subnet mask from a bootstrap server.</p> <p>Denying ICMP address mask request messages can adversely affect diskless systems.</p> <p>Permitting ICMP address mask request messages might allow others to fingerprint the operating system of a host in your network.</p>
ICMP-DEST-UNREACH	3	0	<p>ICMP destination unreachable error message indicates that the destination host is configured to reject the packets.</p> <p>Codes 0, 1, 4, or 5 can be from a gateway. Codes 2 or 3 can be from a host (RFC 792).</p> <p>Denying ICMP destination unreachable error messages can remove the assumption that a host is up and running behind a J Series or an SRX Series device.</p> <p>Permitting ICMP destination unreachable error messages can allow some assumptions, such as security filtering, to be made about the network.</p>
ICMP Fragment Needed	3	4	<p>ICMP fragmentation error message indicates that fragmentation is needed but the don't fragment flag is set.</p> <p>We recommend denying these messages from the Internet to an internal network.</p>
ICMP FragmentReassembly	11	1	<p>ICMP fragment reassembly time exceeded error indicates that a host reassembling a fragmented message ran out of time and dropped the packet. This message is sometimes sent.</p> <p>We recommend denying these messages from the Internet (external) to the trusted (internal) network.</p>

Table 16: ICMP Messages (*continued*)

ICMP Message Name	Type	Code	Description
ICMP-HOST-UNREACH	3	1	<p>ICMP host unreachable error messages indicate that routing table entries do not list or list as infinity a particular host. Sometimes this error is sent by gateways that cannot fragment when a packet requiring fragmentation is received.</p> <p>We recommend denying these messages from the Internet to a trusted network.</p> <p>Permitting these messages allows others to be able to determine your internal hosts IP addresses by a process of elimination or make assumptions about gateways and fragmentation.</p>
ICMP-INFO	15	0	ICMP-INFO query messages allow diskless host systems to query the network and self-configure.
<ul style="list-style-type: none"> Request Reply 	16	0	<p>Denying ICMP address mask request messages can adversely affect diskless systems.</p> <p>Permitting ICMP address mask request messages might allow others to broadcast information queries to a network segment to determine computer type.</p>
ICMP-PARAMETER-PROBLEM	12	0	<p>ICMP parameter problem error messages notify you when incorrect header parameters are present and have caused a packet to be discarded.</p> <p>We recommend denying these messages from the Internet to a trusted network.</p> <p>Permitting ICMP parameter problem error messages allows others to make assumptions about your network.</p>
ICMP-PORT-UNREACH	3	3	<p>ICMP port unreachable error messages indicate that gateways processing datagrams requesting certain ports are unavailable or unsupported in the network.</p> <p>We recommend denying these messages from the Internet to a trusted network.</p> <p>Permitting ICMP port unreachable error messages can allow others to determine which ports you use for certain protocols.</p>

Table 16: ICMP Messages (*continued*)

ICMP Message Name	Type	Code	Description
ICMP-PROTOCOL-UNREACH	3	2	<p>ICMP protocol unreachable error messages indicate that gateways processing datagrams requesting certain protocols are unavailable or unsupported in the network.</p> <p>We recommend denying these messages from the Internet to a trusted network.</p> <p>Permitting ICMP protocol unreachable error messages can allow others to determine what protocols your network is running.</p>
ICMP-REDIRECT	5	0	<p>ICMP redirect network error messages are sent by a J Series or an SRX Series device.</p> <p>We recommend denying these messages from the Internet to a trusted network.</p>
ICMP-REDIRECT-HOST	5	1	<p>ICMP redirect messages indicate datagrams destined for the specified host to be sent along another path.</p>
ICMP-REDIRECT-TOS-HOST	5	3	<p>ICMP redirect type of service (TOS) and host error is a type of message.</p>
ICMP-REDIRECT-TOS-NET	5	2	<p>ICMP redirect TOS and network error is a type of message.</p>
ICMP-SOURCE-QUENCH	4	0	<p>ICMP source quench error message indicates that a device does not have the buffer space available to accept, queue, and send the packets on to the next hop.</p> <p>Denying these messages will not help or impair internal network performance.</p> <p>Permitting these messages can allow others to know that a device is congested, making it a viable attack target.</p>
ICMP-SOURCE-ROUTE-FAIL	3	5	<p>ICMP source route failed error message</p> <p>We recommend denying these messages from the Internet (external).</p>
ICMP-TIME-EXCEEDED	11	0	<p>ICMP time-to-live (TTL) exceeded error message indicates that a packet's TTL setting reached zero before the packet reached its destination. This ensures that older packets are discarded before resent ones are processed.</p> <p>We recommend denying these messages from a trusted network out to the Internet.</p>

Table 16: ICMP Messages (*continued*)

ICMP Message Name	Type	Code	Description
ICMP-TIMESTAMP	13	0	ICMP-TIMESTAMP query messages provide the mechanism to synchronize time and coordinate time distribution in a large, diverse network.
<ul style="list-style-type: none"> Request Reply 	14	0	
Ping (ICMP ECHO)	8	0	<p>Ping is a utility to determine whether a specific host is accessible by its IP address.</p> <p>Denying ping functionality removes your ability to check to see if a host is active.</p> <p>Permitting ping can allow others to execute a denial-of-service (DoS) or Smurf attack.</p>
ICMP-ECHO-FRAGMENT-ASSEMBLY-EXPIRE	11	1	<p>ICMP fragment echo reassembly time expired error message indicates that the reassembly time was exceeded.</p> <p>We recommend denying these messages.</p>
Traceroute	30	0	Traceroute is a utility to indicate the path to access a specific host.
<ul style="list-style-type: none"> Forward Discard 	30	1	We recommend denying this utility from the Internet (external) to your trusted network (internal).

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Default Behaviour of ICMP Unreachable Errors on page 154
 - Example: Configuring Applications and Application Sets on page 141

Default Behaviour of ICMP Unreachable Errors

For different levels of security, the default behavior for ICMP unreachable errors from downstream Juniper Networks device is handled as follows:

- Sessions do not close for ICMP type-3 code-4 messages.
ICMP messages pass through without dropping sessions. Packets are, however, dropped per session.
- Sessions do not close on receiving any kind of ICMP unreachable messages.
- Sessions store ICMP unreachable message, thereby restricting the number of messages flowing through to 1.

One ICMP unreachable message is generated globally per router. The remaining ICMP unreachable errors are dropped.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Understanding the ICMP Predefined Policy Application on page 150
 - Example: Configuring Applications and Application Sets on page 141

Understanding Internet-Related Predefined Policy Applications

When you create a policy, you can specify predefined Internet-related applications for the policy.

Table 17 on page 155 lists Internet-related predefined applications. Depending on your network requirements, you can choose to permit or deny any or all of these applications. Each entry lists the application name, default receiving port, and application description.

Table 17: Predefined Applications

Application Name	Port(s)	Application Description
AOL	5190-5193	America Online Internet service provider (ISP) provides Internet, chat, and instant messaging applications.
DHCP relay	67 (default)	Dynamic Host Configuration Protocol.
DHCP	68 client 67 server	Dynamic Host Configuration Protocol allocates network addresses and delivers configuration parameters from server to hosts.
DNS	53	Domain Name System translates domain names into IP addresses.
FTP	20 data 21 control	File Transfer Protocol (FTP) allows the sending and receiving of files between machines. You can choose to deny or permit ANY (GET or PUT) or to selectively permit or deny either GET or PUT. GET receives files from another machine and PUT sends files to another machine. We recommend denying FTP applications from untrusted sources (Internet).
Gopher	70	Gopher organizes and displays Internet servers' contents as a hierarchically structured list of files. We recommend denying Gopher access to avoid exposing your network structure.
HTTP	8080	HyperText Transfer Protocol is the underlying protocol used by the World Wide Web (WWW). Denying HTTP application disables your users from viewing the Internet. Permitting HTTP application allows your trusted hosts to view the Internet.
HTTP-EXT	—	Hypertext Transfer Protocol with extended nonstandard ports

Table 17: Predefined Applications (*continued*)

Application Name	Port(s)	Application Description
HTTPS	443	<p>Hypertext Transfer Protocol with Secure Sockets Layer (SSL) is a protocol for transmitting private documents through the Internet.</p> <p>Denying HTTPS disables your users from shopping on the Internet and from accessing certain online resources that require secure password exchange.</p> <p>Permitting HTTPS allows your trusted hosts to participate in password exchange, shop online, and visit various protected online resources that require user login.</p>
Internet Locator Service	—	Internet Locator Service includes LDAP, User Locator Service, and LDAP over TSL/SSL.
IRC	6665-6669	Internet Relay Chat (IRC) allows people connected to the Internet to join live discussions.
LDAP	389	Lightweight Directory Access Protocol is a set of protocols used to access information directories.
PC-Anywhere	—	PC-Anywhere is a remote control and file transfer software.
TFTP	69	Trivial File transfer Protocol (TFTP) is a protocol for simple file transfer.
WAIS	—	Wide Area Information Server is a program that finds documents on the Internet.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Understanding Dynamic Routing Protocols Predefined Policy Applications on page 157
 - Example: Configuring Applications and Application Sets on page 141

Understanding Microsoft Predefined Policy Applications

When you create a policy, you can specify predefined Microsoft applications for the policy.

Table 18 on page 156 lists predefined Microsoft applications, parameters associated with each application, and a brief description of each application. Parameters include universal unique identifiers (UUIDs) and TCP/UDP source and destination ports. A UUID is a 128-bit unique number generated from a hardware address, a timestamp, and seed values.

Table 18: Predefined Microsoft Applications

Application	Parameter/UUID	Description
Junos MS-RPC-EPM	135 e1af8308-5d1f-11c9-91a4-08002b14a0fa	Microsoft remote procedure call (RPC) Endpoint Mapper (EPM) Protocol.

Table 18: Predefined Microsoft Applications (*continued*)

Application	Parameter/UUID	Description
Junos MS-RPC	—	Any Microsoft remote procedure call (RPC) applications.
Junos MS-RPC-MSEXCHANGE	3 members	Microsoft Exchange application group includes: <ul style="list-style-type: none"> Junos-MS-RPC-MSEXCHANGE-DATABASE Junos-MS-RPC-MSEXCHANGE-DIRECTORY Junos-MS-RPC-MSEXCHANGE-INFO-STORE
Junos-MS-RPC-MSEXCHANGE-DATABASE	1a190310-bb9c-11cd-90f8-00aa00466520	Microsoft Exchange Database application.
Junos-MS-RPC-MSEXCHANGE-DIRECTORY	f5cc5a18-4264-101a-8c59-08002b2f8426 f5cc5a7c-4264-101a-8c59-08002b2f8426 f5cc59b4-4264-101a-8c59-08002b2f8426	Microsoft Exchange Directory application.
Junos-MS-RPC-MSEXCHANGE-INFO-STORE	0e4a0156-dd5d-11d2-8c2f-00c04fb6bcde 1453c42c-0fa6-11d2-a910-00c04f990f3b 10f24e8e-0fa6-11d2-a910-00c04f990f3b 1544f5e0-613c-11d1-93df-00c04fd7bd09	Microsoft Exchange Information Store application.
Junos-MS-RPC-TCP	—	Microsoft Transmission Control Protocol (TCP) application.
Junos-MS-RPC-UDP	—	Microsoft User Datagram Protocol (UDP) application.
Junos-MS-SQL	—	Microsoft Structured Query Language (SQL).
Junos-MSN	—	Microsoft Network Messenger application.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Example: Configuring Applications and Application Sets on page 141

Understanding Dynamic Routing Protocols Predefined Policy Applications

When you create a policy, you can specify predefined dynamic routing protocol applications for the policy.

Depending on your network requirements, you can choose to permit or deny messages generated from these dynamic routing protocols and packets of these dynamic routing protocols. Table 19 on page 158 lists each supported dynamic routing protocol by name, port, and description.

Table 19: Dynamic Routing Protocols

Dynamic Routing Protocol	Port	Description
RIP	520	RIP is a common distance-vector routing protocol.
OSPF	89	OSPF is a common link-state routing protocol.
BGP	179	BGP is an exterior/interdomain routing protocol.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Example: Configuring Applications and Application Sets on page 141

Understanding Streaming Video Predefined Policy Applications

When you create a policy, you can specify predefined streaming video applications for the policy.

Table 20 on page 158 lists each supported streaming video application by name and includes the default port and description. Depending on your network requirements, you can choose to permit or deny any or all of these applications.

Table 20: Supported Streaming Video Applications

Application	Port	Description
H.323	TCP source 1-65535; TCP destination 1720, 1503, 389, 522, 1731 UDP source 1-65535; UDP source 1719	H.323 is a standard approved by the International Telecommunication Union (ITU) that defines how audiovisual conference data is transmitted across networks.
NetMeeting	TCP source 1-65535; TCP destination 1720, 1503, 389, 522 UDP source 1719	Microsoft NetMeeting uses TCP to provide teleconferencing (video and audio) applications over the Internet.
Real media	TCP source 1-65535; TCP destination 7070	Real Media is streaming video and audio technology.
RTSP	554	Real-Time Streaming Protocol (RTSP) is for streaming media applications
SIP	5056	Session Initiation Protocol (SIP) is an Application-Layer control protocol for creating, modifying, and terminating sessions.

Table 20: Supported Streaming Video Applications (*continued*)

Application	Port	Description
VDO Live	TCP source 1-65535; TCP destination 7000-7010	VDOLive is a scalable, video streaming technology.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Example: Configuring Applications and Application Sets on page 141

Understanding Sun RPC Predefined Policy Applications

When you create a policy, you can specify predefined Sun RPC applications for the policy.

Table 21 on page 159 lists each Sun remote procedure call Application Layer Gateway (RPC ALG) application name, parameters, and full name.

Table 21: RPC ALG Applications

Application	Program Numbers	Full Name
SUN-RPC-PORTMAPPER	111100000	Sun RPC Portmapper protocol
SUN-RPC-ANY	ANY	Any Sun RPC applications
SUN-RPC-PROGRAM-MOUNTD	100005	Sun RPC Mount Daemon
SUN-RPC-PROGRAM-NFS	100003 100227	Sun RPC Network File System
SUN-RPC-PROGRAM-NLOCKMGR	100021	Sun RPC Network Lock Manager
SUN-RPC-PROGRAM-RQUOTAD	100011	Sun RPC Remote Quota Daemon
SUN-RPC-PROGRAM-RSTATD	100001	Sun RPC Remote Status Daemon
SUN-RPC-PROGRAM-RUSERD	100002	Sun RPC Remote User Daemon
SUN-RPC-PROGRAM-SADMIND	100232	Sun RPC System Administration Daemon
SUN-RPC-PROGRAM-SPRAYD	100012	Sun RPC Spray Daemon
SUN-RPC-PROGRAM-STATUS	100024	Sun RPC Status
SUN-RPC-PROGRAM-WALLD	100008	Sun RPC Wall Daemon
SUN-RPC-PROGRAM-YPBIND	100007	SUN RPC Yellow Page Bind application

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Example: Configuring Applications and Application Sets on page 141

Understanding Security and Tunnel Predefined Policy Applications

When you create a policy, you can specify predefined security and tunnel applications for the policy.

Table 22 on page 160 lists each supported application and gives the default port(s) and a description of each entry.

Table 22: Supported Applications

Application	Port	Description
IKE	UDP source 1-65535; UDP destination 500 4500 (used for NAT traversal)	<p>Internet Key protocol (IKE) is a protocol to obtain authenticated keying material for use with ISAKMP.</p> <p>When configuring auto IKE, you can choose from three predefined Phase 1 or Phase 2 proposals:</p> <ul style="list-style-type: none"> • Standard: AES and 3DES • Basic: DES and two different types of authentication algorithms • Compatible: Four commonly used authentication and encryption algorithms
L2TP	1723	L2TP combines PPTP with Layer 2 Forwarding (L2F) for remote access.
PPTP	—	Point-to-Point Tunneling Protocol allows corporations to extend their own private network through private <i>tunnels</i> over the public Internet.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Example: Configuring Applications and Application Sets on page 141

Understanding IP-Related Predefined Policy Applications

When you create a policy, you can specify predefined IP-related applications for the policy.

Table 23 on page 161 lists the predefined IP-related applications. Each entry includes the default port and a description of the application.

Table 23: Predefined IP-Related Applications

Application	Port	Description
Any	—	Any application
TCP-ANY	1-65535	Any protocol using the TCP TCPMUX port 1
UDP-ANY	137	Any protocol using the UDP

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Example: Configuring Applications and Application Sets on page 141

Understanding Instant Messaging Predefined Policy Applications

When you create a policy, you can specify predefined instant messaging applications for the policy.

Table 24 on page 161 lists predefined Internet-messaging applications. Each entry includes the name of the application, the default or assigned port, and a description of the application.

Table 24: Predefined Internet-Messaging Applications

Application	Port	Description
Gnutella	6346 (default)	Gnutella is a public domain file sharing protocol that operates over a distributed network. You can assign any port, but the default is 6346.
MSN	1863	Microsoft Network Messenger is a utility that allows you to send instant messages and talk online.
NNTP	119	Network News Transport Protocol is a protocol used to post, distribute, and retrieve USENET messages.
SMB	445	Server Message Block (SMB) over IP is a protocol that allows you to read and write files to a server on a network.
YMSG	5010	Yahoo! Messenger is a utility that allows you to check when others are online, send instant messages, and talk online.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Understanding Management Predefined Policy Applications on page 162
 - Example: Configuring Applications and Application Sets on page 141

Understanding Management Predefined Policy Applications

When you create a policy, you can specify predefined management applications for the policy.

Table 25 on page 162 lists the predefined management applications. Each entry includes the name of the application, the default or assigned port, and a description of the application.

Table 25: Predefined Management Applications

Application	Port	Description
NBNAME	137	NetBIOS Name application displays all NetBIOS name packets sent on UDP port 137.
NDBDS	138	NetBIOS Datagram application, published by IBM, provides connectionless (datagram) applications to PCs connected with a broadcast medium to locate resources, initiate sessions, and terminate sessions. It is unreliable and the packets are not sequenced.
NFS	—	Network File System uses UDP to allow network users to access shared files stored on computers of different types. SUN RPC is a building block of NFS.
NS Global	—	NS-Global is the central management protocol for Juniper Networks Firewall/VPN devices.
NS Global PRO	—	NS Global-PRO is the scalable monitoring system for the Juniper Networks Firewall/VPN device family.
NSM	—	Network and Security Manager
NTP	123	Network Time Protocol provides a way for computers to synchronize to a time reference.
RLOGIN	513	RLOGIN starts a terminal session on a remote host.
RSH	514	RSH executes a shell command on a remote host.
SNMP	161	Simple Network Management Protocol is a set of protocols for managing complex networks.
SQL*Net V1	66	SQL*Net Version 1 is a database language that allows for the creation, access, modification, and protection of data.
SQL*Net V2	66	SQL*Net Version 2 is a database language that allows for the creation, access, modification, and protection of data.
MSSQL	1433 (default instance)	Microsoft SQL is a proprietary database server tool that allows for the creation, access, modification, and protection of data.

Table 25: Predefined Management Applications (*continued*)

Application	Port	Description
SSH	22	SSH is a program to log into another computer over a network through strong authentication and secure communications on an unsecure channel.
SYSLOG	514	Syslog is a UNIX program that sends messages to the system logger.
Talk	517-518	Talk is a visual communication program that copies lines from your terminal to that of another user.
Telnet	23	Telnet is a UNIX program that provides a standard method of interfacing terminal devices and terminal-oriented processes to each other.
WinFrame	—	WinFrame is a technology that allows users on non-Windows machines to run Windows applications.
X-Windows	—	X-Windows is the windowing and graphics system that Motif and OpenLook are based on.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Example: Configuring Applications and Application Sets on page 141

Understanding Mail Predefined Policy Applications

When you create a policy, you can specify predefined mail applications for the policy.

Table 26 on page 163 lists the predefined mail applications. Each includes the name of the application, the default or assigned port number, and a description of the application.

Table 26: Predefined Mail Applications

Application	Port	Description
IMAP	143	Internet Message Access Protocol is used for retrieving messages.
Mail (SMTP)	25	Simple Mail Transfer Protocol is used to send messages between servers.
POP3	110	Post Office Protocol is used for retrieving e-mail.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Example: Configuring Applications and Application Sets on page 141

Understanding UNIX Predefined Policy Applications

When you create a policy, you can specify predefined UNIX applications for the policy.

Table 27 on page 164 lists the predefined UNIX applications. Each entry includes the name of the application, the default or assigned port, and a description of the application.

Table 27: Predefined UNIX Applications

Application	Port	Description
FINGER	79	Finger is a UNIX program that provides information about the users.
UUCP	117	UNIX-to-UNIX Copy Protocol (UUCP) is a UNIX utility that enables file transfers between two computers over a direct serial or modem connection.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Example: Configuring Applications and Application Sets on page 141

Understanding Miscellaneous Predefined Policy Applications

When you create a policy, you can specify miscellaneous predefined applications for the policy.

Table 28 on page 164 lists predefined miscellaneous applications. Each entry includes the application name, default or assigned port, and a description of the application.

Table 28: Predefined Miscellaneous Applications

Application	Port	Description
CHARGEN	19	Character Generator Protocol is a UDP- or TCP-based debugging and measurement tool.
DISCARD	9	Discard protocol is an Application Layer protocol that describes a process for discarding TCP or UDP data sent to port 9.
IDENT	113	Identification protocol is a TCP/IP Application Layer protocol used for TCP client authentication.
LPR	515 listen; 721-731 source range (inclusive)	Line Printer Daemon protocol is a TCP-based protocol used for printing applications.
RADIUS	1812	Remote Authentication Dial-In User Service application is a server program used for authentication and accounting purposes.

Table 28: Predefined Miscellaneous Applications (*continued*)

Application	Port	Description
SQLMON	1434 (SQL Monitor Port)	SQL monitor (Microsoft)
VNC	5800	Virtual Network Computing facilitates viewing and interacting with another computer or mobile Juniper Networks device connected to the Internet.
WHOIS	43	Network Directory Application Protocol is a way to look up domain names.
IPsec-NAT	—	IPSEC-NAT allows Network Address Translation for ISAKMP and ESP packets.
SCCP	2000	Cisco Station Call Control Protocol (SCCP) uses the signaling connection control port to provide high availability and flow control.
VoIP	—	Voice over IP application group provides voice applications over the Internet and includes H.323 and Session Initiation Protocol (SIP).

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policy Applications Overview on page 139
 - Example: Configuring Applications and Application Sets on page 141

PART 4

Application Layer Gateways

- ALGs on page 169
- H.323 ALGs on page 173
- ALG for IKE and ESP on page 199
- SIP ALGs on page 207
- SCCP ALGs on page 245
- MGCP ALGs on page 261
- RPC ALGs on page 287

CHAPTER 9

ALGs

- ALG Overview on page 169
- Understanding ALG Types on page 170

ALG Overview

An *Application Layer Gateway (ALG)* is a software component that is designed to manage specific protocols such as Session Initiation Protocol (SIP) or FTP on Juniper Networks devices running Junos OS. The ALG module is responsible for Application-Layer aware packet processing.

ALG functionality can be triggered either by a service or application configured in the security policy:

- A *service* is an object that identifies an application protocol using Layer 4 information (such as standard and accepted TCP and UDP port numbers) for an application service (such as Telnet, FTP, SMTP, and HTTP).
- An *application* specifies the Layer 7 application that maps to a Layer 4 service.

A predefined service already has a mapping to a Layer 7 application. However, for custom services, you must link the service to an application explicitly, especially if you want the policy to apply an ALG.

ALGs for packets destined to well-known ports are triggered by service type. The ALG intercepts and analyzes the specified traffic, allocates resources, and defines dynamic policies to permit the traffic to pass securely through the device:

1. When a packet arrives at the device, the flow module forwards the packet according to the security rule set in the policy.
2. If a policy is found to permit the packet, the associated service type or application type is assigned and a session is created for this type of traffic.
3. If a session is found for the packet, no policy rule match is needed. The ALG module is triggered if that particular service or application type requires the supported ALG processing.

The ALG also inspects the packet for embedded IP address and port information in the packet payload, and performs Network Address Translation (NAT) processing if necessary.

The ALG also opens a gate for the IP address and port number to permit data exchange for the session. The control session and data session can be coupled to have the same timeout value, or they can be independent.

ALGs are supported on chassis clusters. For information about chassis clusters, see “Chassis Cluster Overview” on page 795.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding ALG Types on page 170
 - Understanding H.323 ALGs on page 173
 - Understanding SIP ALGs on page 207
 - Understanding SCCP ALGs on page 245
 - Understanding MGCP ALGs on page 261
 - Understanding RPC ALGs on page 287

Understanding ALG Types

Junos OS supports voice-over-IP Application Layer Gateways (VoIP ALGs) and basic data ALGs. (Note that supported ALG types vary depending on which hardware device you are using.)

VoIP ALGs provide stateful Application Layer inspection and Network Address Translation (NAT) capabilities to VoIP signaling and media traffic. The ALG inspects the state of transactions, or calls, and forwards or drops packets based on those states.

Junos OS supports the following VoIP ALGs:

- **H.323**—The H.323 ALG provides support for the H.323 legacy VoIP protocol. The ALG lets you secure VoIP communication between terminal hosts, such as IP phones and multimedia devices. In such a telephony system, the gatekeeper device manages call registration, admission, and call status for VoIP calls. Gatekeepers can reside in the two different zones or in the same zone.
- **SIP**—The SIP ALG provides support for the Session Initiation Protocol (SIP). SIP is an Internet Engineering Task Force (IETF)-standard protocol for initiating, modifying, and terminating multimedia sessions over the Internet. Such sessions might include conferencing, telephony, or multimedia, with features such as instant messaging and application-level mobility in network environments.
- **SCCP**—The SCCP ALG provides support for Skinny Client Control Protocol (SCCP). SCCP is a Cisco proprietary protocol for call signaling. Skinny is based on a call-agent-based call-control architecture. The control protocol uses binary-coded frames encoded on TCP frames sent to well-known TCP port number destinations to set up and tear down RTP media sessions.
- **MGCP**—The MGCP ALG provides support for Media Gateway Control Protocol (MGCP). MGCP is a text-based Application Layer protocol used for call setup and call control between the media gateway and the media gateway controller (MGC).

Junos OS also supports the following data ALGs:

- **DNS**—Provides an ALG for the Domain Name System. The DNS ALG monitors DNS query and reply packets and closes session if the DNS flag indicates the packet is a reply message.
- **FTP**—Provides an ALG for the File Transfer Protocol (FTP). The FTP ALG monitors PORT, PASV, and 227 commands. It performs NAT on the IP, port, or both in the message and gate opening on the device as necessary. The FTP ALG supports FTP put and FTP get command blocking. When the **FTP_NO_PUT** or **FTP_NO_GET** command is set in the policy, the FTP ALG sends back a blocking command and closes the associated opened gate when the **FTP STOR** or **FTP RETR** command is observed.
- **TFTP**—Provides an ALG for the Trivial File Transfer Protocol (TFTP). The TFTP ALG processes TFTP packets that initiate the request and opens a gate to allow return packets from the reverse direction to the port that sends the request.
- **PPTP**—Provides an ALG for the Point-to-Point Tunneling Protocol (PPTP). The PPTP is a Layer 2 protocol that tunnels PPP data across TCP/IP networks. The PPTP client is freely available on Windows systems and is widely deployed for building Virtual Private Networks (VPNs).
- **REAL**—Provides an ALG for the Real-Time Streaming Protocol.
- **MSRPC**—Provides an ALG for the Microsoft Remote Procedure Call.
- **SUNRPC**—Provides an ALG for the SUN Remote Procedure Call.
- **RSH**—Provides an ALG for the Remote Shell (RSH). The RSH ALG handles TCP packets destined for port 514 and processes the RSH port command. The RSH ALG performs NAT on the port in the port command and opens gates as necessary.
- **SQL**—Provides an ALG for the Structured Query Language (SQL). The SQLNET ALG processes SQL TNS response frame from the server side. It parses the packet and looks for the (HOST=ipaddress), (PORT=port) pattern and performs NAT and gate opening on the client side for the TCP data channel.
- **TALK**—Provides an ALG for the TALK Protocol. The TALK protocol uses UDP port 517 and port 518 for control channel connections. The talk program consists of a server and a client. The server handles client notifications and helps to establish talk sessions. There are two types of talk servers: *ntalk* and *talkd*. The TALK ALG processes packets of both *ntalk* and *talkd* formats. It also performs NAT and gate opening as necessary.

For information about enabling and configuring each of these ALGs through J-Web, select the **Configure>Security>ALG** page in the J-Web user interface and click **Help**.

Related Topics • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- ALG Overview on page 169
- Understanding H.323 ALGs on page 173
- Understanding SIP ALGs on page 207
- Understanding SCCP ALGs on page 245
- Understanding MGCP ALGs on page 261

- [Understanding RPC ALGs on page 287](#)

CHAPTER 10

H.323 ALGs

- Understanding H.323 ALGs on page 173
- Understanding the Avaya H.323 ALG on page 175
- H.323 ALG Configuration Overview on page 176
- H.323 ALG Endpoint Registration Timeouts on page 177
- H.323 ALG Media Source Port Ranges on page 179
- H.323 ALG DoS Attack Protection on page 180
- H.323 ALG Unknown Message Types on page 182
- Example: Passing H.323 ALG Traffic to a Gatekeeper in the Internal Zone on page 184
- Example: Passing H.323 ALG Traffic to a Gatekeeper in the External Zone on page 189
- Example: Using NAT and the H.323 ALG to Enable Incoming Calls (CLI) on page 194
- Example: Using NAT and the H.323 ALG to Enable Outgoing Calls (CLI) on page 196

Understanding H.323 ALGs

The H.323 standard is a legacy voice-over-IP (VoIP) protocol defined by the International Telecommunication Union (ITU-T). H.323 consists of a suite of protocols (such as H.225.0 and H.245) that are used for call signaling and call control for VoIP.

H.323 uses the ASN.1 coding format. It sets up the dynamic links for data, video, and audio streams, following the protocols Q.931 (with port number 1720) and H.245. There are three major processes in H.323:

- Gatekeeper Discovery—An endpoint finds its gatekeeper through the gatekeeper discovery process, through broadcast or unicast (to a known IP and the well-known UDP port 1719). (Junos OS supports unicast only.)
- Endpoint Registration, Admission, and Status—An endpoint registers to a gatekeeper and asks for its management. Before making a call, an endpoint asks its gatekeeper for permission to place the call. In both registration and admission phases, the Registration, Admission, and Status (RAS) channel is used. The Transport Service Access Point (TSAP) can be either the well-known UDP port (1719) or a dynamically assigned port from the discovery or registration phase.
- Call Control and Call Setup—Calls can be established within a zone or across two zones, or even across multiple zones (multipoint conference). The call setup and tear

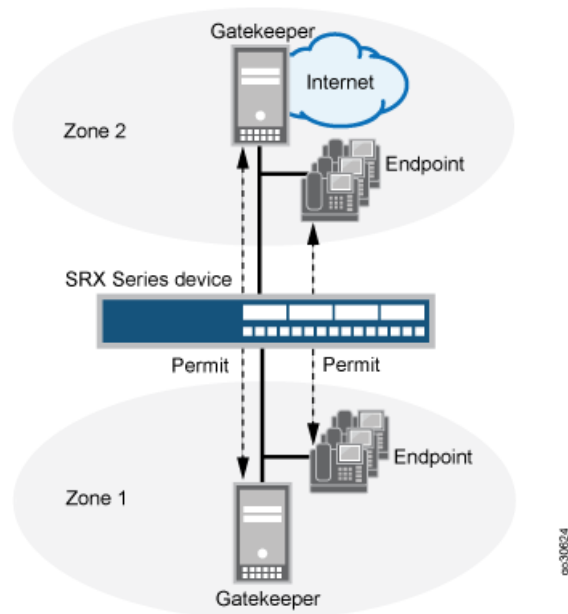
down is performed through the call signaling channel whose TSAP is the well-known TCP port (1720). The call control, including opening/closing media channels between two endpoints, is performed through the call control channel whose TSAP is dynamically assigned from the previous call signaling process. H.245 messages are used in the call control channel, and are encoded using ASN.1.



NOTE: Detailed information on H.323 can be found in ITU-T Recommendation H.323.

The H.323 Application Layer Gateway (ALG) lets you secure VoIP communication between terminal hosts, such as IP phones and multimedia devices. In such a telephony system, the gatekeeper device manages call registration, admission, and call status for VoIP calls. Gatekeepers can reside in the two different zones or in the same zone. (See Figure 13 on page 174.)

Figure 13: H.323 ALG for VoIP Calls



NOTE: The illustration uses IP phones for illustrative purposes, although it is possible to make configurations for other hosts that use VoIP, such as Microsoft NetMeeting multimedia devices.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- ALG Overview on page 169
- Understanding the Avaya H.323 ALG on page 175
- H.323 ALG Configuration Overview on page 176

Understanding the Avaya H.323 ALG

The H.323 standard is a legacy voice-over-IP (VoIP) protocol defined by the International Telecommunication Union (ITU-T). H.323 consists of a suite of protocols (such as H.225.0 and H.245) that are used for call signaling and call control for VoIP. The processes for configuring the H.323 standard Application Layer Gateway (ALG) and the proprietary Avaya H.323 ALG are the same.

However, Avaya H.323 ALG has some special features. To understand and configure the Avaya H.323-specific features listed here, see the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

This topic contains the following sections:

- Avaya H.323 ALG-Specific Features on page 175
- Call Flow Details in the Avaya H.323 ALG on page 175

Avaya H.323 ALG-Specific Features

Avaya H.323-specific features are as follows:

- H.323 Fast Connect
- H.323 asymmetric media
- Call waiting
- Call forwarding
- Voice mail
- Call identification
- Conference calling

Call Flow Details in the Avaya H.323 ALG

- Connecting the Phone into the Network—Avaya performs the Q.931 Setup/Connect negotiation when the phone is wired into the network rather than when a call is being initiated.
- Making a call—When a call is made, because the PBX has already stored the capabilities for each phone when the phone is connected to the network, no further Q.931 and PBX negotiations are required to set up the call. It no longer exchanges Q.931 Setup and Connect messages with the PBX. The phone and the PBX exchange H.323 Facility messages to set up the call.
- Registering with a CM—When a call has been made, Avaya H.323 registers with the Avaya Communication Manager (CM). The registration process is similar to a generic H.323 standard registration process.



NOTE: The direct mode and tunnel mode are not defined by Avaya H.323 ALG.

For a call to work, the CM must be deployed with Avaya Endpoints. During the call, RAS and Q.931 messages are exchanged between the CM and the Avaya Endpoints.



NOTE: For Avaya H.323 with a source Network Address Translation (NAT) pool, the registration process allows only one IP address in the pool.

- Setting up Real-Time Transport Protocol (RTP)/Real-Time Control Protocol (RTCP) ports—The Q.931 Setup, Facility and Information messages are used to set up RTP/RTCP ports. The hierarchy for an Avaya H.323 session is Q.931, RTP/RTCP, Parent, and then Child.



NOTE: H.245 ports are not used in an Avaya call flow process.

- Using Avaya H.323 counters—The counters for calls and active calls are not applicable to the Avaya H.323 ALG. The call creation and tearing down is done by Facility messages afterward. When resources are allocated for a call, all counters for calls and active calls increment. If resources are allocated for a call multiple times, messages belonging to the same call that pass the firewall multiple times will trigger multiple increments of the counters. In other words, messages that belong to the same call and pass the firewall multiple times might trigger multiple increments of the counters if the resource for a call needs to be allocated multiple times.

For example, in the two-zone case, the setup and connect message pair allocates one call resource. The active call counter is increased once. Each time the setup and connect message pair passes the firewall, a different call resource with unique interfaces and NAT is allocated. Therefore, the counter increments twice in a three-zone scenario.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- ALG Overview on page 169
- Understanding H.323 ALGs on page 173
- H.323 ALG Configuration Overview on page 176

H.323 ALG Configuration Overview

The H.323 Application Layer Gateway (ALG) is enabled by default on the device—no action is required to enable it. However, you might choose to fine-tune H.323 ALG operations by using the following instructions:

1. Specify how long an endpoint registration entry remains in the Network Address Translation (NAT) table. For instructions, see “Example: Setting H.323 ALG Endpoint Registration Timeouts” on page 177.
2. Enable media traffic on a narrow or wide range of ports. For instructions, see “Example: Setting H.323 ALG Media Source Port Ranges” on page 179.

3. Protect the H.323 gatekeeper from denial-of-service (DoS) flood attacks. For instructions, see “Example: Configuring H.323 ALG DoS Attack Protection” on page 181.
4. Enable unknown messages to pass when the session is in NAT mode and route mode. For instructions, see “Example: Allowing Unknown H.323 ALG Message Types” on page 183.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding H.323 ALGs on page 173
 - Example: Passing H.323 ALG Traffic to a Gatekeeper in the Private Zone on page 184
 - Example: Passing H.323 ALG Traffic to a Gatekeeper in the External Zone on page 189
 - Example: Using NAT and the H.323 ALG to Enable Incoming Calls (CLI) on page 194
 - Example: Using NAT and the H.323 ALG to Enable Outgoing Calls (CLI) on page 196

H.323 ALG Endpoint Registration Timeouts

- Understanding H.323 ALG Endpoint Registration Timeouts on page 177
- Example: Setting H.323 ALG Endpoint Registration Timeouts on page 177

Understanding H.323 ALG Endpoint Registration Timeouts

In Network Address Translation (NAT) mode, when endpoints in the protected network behind the Juniper Networks device register with the H.323 gatekeeper, the device adds an entry to the NAT table containing a mapping of the public-to-private address for each endpoint. These entries make it possible for endpoints in the protected network to receive incoming calls.

You set an endpoint registration timeout to specify how long an endpoint registration entry remains in the NAT table. To ensure uninterrupted incoming call service, set the endpoint registration timeout to a value equal to or greater than the keepalive value the administrator configures on the gatekeeper. The range is 10 to 50,000 seconds, the default value is 3600 seconds.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding H.323 ALGs on page 173
 - H.323 ALG Configuration Overview on page 176
 - Example: Setting H.323 ALG Endpoint Registration Timeouts on page 177

Example: Setting H.323 ALG Endpoint Registration Timeouts

This example shows how to specify the endpoint registration timeout.

- Requirements on page 178
- Overview on page 178

- Configuration on page 178
- Verification on page 178

Requirements

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

Overview

You set an endpoint registration timeout range to specify how long an endpoint registration entry remains in the NAT table. The range is 10 to 50,000 seconds, and the default value is 3600 seconds.

Configuration

J-Web Quick Configuration

To specify the H.323 ALG endpoint registration timeout:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. In the Timeout for endpoints box, type **5000**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To specify the H.323 ALG endpoint registration timeout:

1. Configure the H.323 ALG and set the endpoint registration timeout to 5000 seconds.
[edit]
user@host# **set security alg h323 endpoint-registration-timeout 5000**
2. If you are done configuring the device, commit the configuration.
[edit]
user@host# **commit**

Verification

To verify the configuration is working properly, enter the **show security alg h323** command and the **show security alg h323 counters** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding H.323 ALG Endpoint Registration Timeouts on page 177
- H.323 ALG Configuration Overview on page 176

H.323 ALG Media Source Port Ranges

- Understanding H.323 ALG Media Source Port Ranges on page 179
- Example: Setting H.323 ALG Media Source Port Ranges on page 179

Understanding H.323 ALG Media Source Port Ranges

The media source port feature enables you to configure the device to allow media traffic on a narrow or wide range of ports. By default, the device listens for H.323 traffic on a wide range of ports. If your endpoint equipment allows you to specify a sending port and a listening port, you might want to narrow the range of ports the device allows media traffic on. This enhances security by opening a smaller pinhole for H.323 traffic.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding H.323 ALGs on page 173
 - H.323 ALG Configuration Overview on page 176
 - Example: Setting H.323 ALG Media Source Port Ranges on page 179

Example: Setting H.323 ALG Media Source Port Ranges

This example shows how to disable the media source port feature.

- Requirements on page 179
- Overview on page 179
- Configuration on page 179
- Verification on page 180

Requirements

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

Overview

The media source port feature enables you to configure the device to allow media traffic on a narrow or wide range of ports. By default, the device listens for H.323 traffic on a narrow range of ports. This example shows how to configure the device to open a wide gate for media traffic by enabling the media source port feature.

Configuration

J-Web Quick Configuration

To disable the media source port feature:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. Select the **Enable Permit media from any source port** check box.

4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

**Step-by-Step
Procedure**

To disable the media source port feature:

1. Set a narrow gate for media traffic by disabling the media source port for the H.323 ALG.

```
[edit]  
user@host# delete security alg h323 media-source-port-any
```

2. If you are done configuring the device, commit the configuration.

```
[edit]  
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg h323** command and the **show security alg h323 counters** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding H.323 ALG Media Source Port Ranges on page 179
- H.323 ALG Configuration Overview on page 176

H.323 ALG DoS Attack Protection

- Understanding H.323 ALG DoS Attack Protection on page 180
- Example: Configuring H.323 ALG DoS Attack Protection on page 181

Understanding H.323 ALG DoS Attack Protection

You can protect the H.323 gatekeeper from denial-of-service (DoS) flood attacks by limiting the number of Registration, Admission, and Status (RAS) messages per second it will attempt to process. Incoming RAS request messages exceeding the threshold you specify are dropped by H.323 Application Layer Gateway (ALG). The range is 2 to 50,000 messages per second, the default value is 1000.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding H.323 ALGs on page 173
- H.323 ALG Configuration Overview on page 176
- Example: Configuring H.323 ALG DoS Attack Protection on page 181

Example: Configuring H.323 ALG DoS Attack Protection

This example shows how to configure the H.323 ALG DoS attack protection feature.

- Requirements on page 181
- Overview on page 181
- Configuration on page 181
- Verification on page 182

Requirements

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

Overview

You can protect the H.323 gatekeeper from DoS flood attacks by limiting the range of Registration, Admission, and Status (RAS) messages per second it will attempt to process. The range is 2 to 50,000 messages per second, and the default value is 1000. This example limits the number of incoming RAS request messages to 5000 messages per second.

Configuration

J-Web Quick Configuration

To configure the H.323 ALG DoS attack protection feature:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. In the Message flood gatekeeper threshold box, type **5000**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure the H.323 ALG DoS attack protection feature:

1. Configure the gatekeeper for the H.323 ALG and set the threshold.

```
[edit]
user@host# set security alg h323 application-screen message-flood gatekeeper
threshold 5000
```
2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg h323** command and the **show security alg h323 counters** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding H.323 ALG DoS Attack Protection on page 180
 - H.323 ALG Configuration Overview on page 176

H.323 ALG Unknown Message Types

- Understanding H.323 ALG Unknown Message Types on page 182
- Example: Allowing Unknown H.323 ALG Message Types on page 183

Understanding H.323 ALG Unknown Message Types

This feature enables you to specify how unidentified H.323 messages are handled by the device. The default is to drop unknown (unsupported) messages.

You can protect the H.323 gatekeeper from denial-of-service (DoS) flood attacks by limiting the number of Registration, Admission, and Status (RAS) messages per second it will attempt to process. Incoming RAS request messages exceeding the threshold you specify are dropped by the H.323 Application Layer Gateway (ALG). The range is 2 to 50,000 messages per second, the default value is 1000.

We do not recommend permitting unknown messages because they can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment. Permitting unknown H.323 messages can help you get your network operational, so that you can analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped. The unknown H.323 message type feature enables you to configure the device to accept H.323 traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.



NOTE: This option applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding H.323 ALGs on page 173
 - H.323 ALG Configuration Overview on page 176
 - Example: Allowing Unknown H.323 ALG Message Types on page 183

Example: Allowing Unknown H.323 ALG Message Types

This example shows how to configure the device to allow unknown H.323 message types in both route and NAT modes.

- Requirements on page 183
- Overview on page 183
- Configuration on page 183
- Verification on page 184

Requirements

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

Overview

This feature enables you to specify how unidentified H.323 messages are handled by the device. The default is to drop unknown (unsupported) messages. The Enable Permit NAT applied option and the **permit-nat-applied** configuration statement specify that unknown messages be allowed to pass if the session is in NAT mode. The Enable Permit routed option and the **permit-routed** configuration statement specify that unknown messages be allowed to pass if the session is in route mode. (Sessions in transparent mode are treated as route mode.)

Configuration

J-Web Quick Configuration

To configure the device to allow unknown H.323 message types in both route and NAT modes:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure the device to allow unknown H.323 message types in both route and NAT modes:

1. Specify that unknown messages be allowed to pass if the session is in NAT mode.

```
[edit]
user@host# set security alg h323 application-screen unknown-message
permit-nat-applied
```
2. Specify that unknown messages be allowed to pass if the session is in route mode.

```
[edit]
user@host# set security alg h323 application-screen unknown-message
permit-routed
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg h323** command and the **show security alg h323 counters** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding H.323 ALG Unknown Message Types on page 182
 - H.323 ALG Configuration Overview on page 176

Example: Passing H.323 ALG Traffic to a Gatekeeper in the Internal Zone

This example shows how to set up two policies that allow H.323 traffic to pass between IP phone hosts and a gatekeeper in the private zone, and an IP phone host (2.2.2.5/32) in the public zone.

- Requirements on page 184
- Overview on page 184
- Configuration on page 185
- Verification on page 187

Requirements

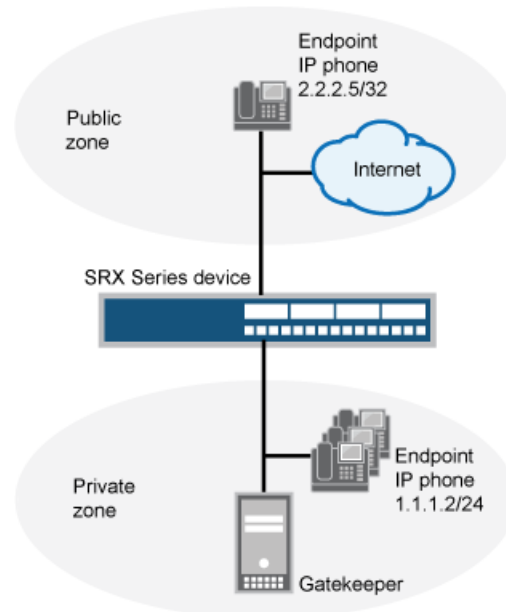
Before you begin:

- Understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.
- Configure security zones. See “Understanding Security Zones” on page 87.

Overview

This example shows how to set up two policies that allow H.323 traffic to pass between IP phone hosts and a gatekeeper in the private zone, and an IP phone host (2.2.2.5/32) in the public zone. The connection to the device can either be with or without NAT. See Figure 14 on page 185.

Figure 14: H.323 Gatekeeper in Zone1



Configuration

CLI Quick Configuration To quickly configure the device to pass H.323 ALG traffic to a gatekeeper in the internal zone, copy the following commands and paste them into the CLI:

```
[edit]
set security zones security-zone public address-book address ip_phone 2.2.2.5/32
set security zones security-zone private address-book address gateway 2.2.2.5/32
set security policies from-zone private to-zone public policy P1 match source-address
any
set security policies from-zone private to-zone public policy P1 match destination-address
IP_Phone
set security policies from-zone private to-zone public policy P1 match application
junos-h323
set security policies from-zone private to-zone public policy P1 then permit
set security policies from-zone public to-zone private policy P2 match source-address
any
set security policies from-zone public to-zone private policy P2 match destination-address
gateway
set security policies from-zone public to-zone private policy P2 match application
junos-h323
set security policies from-zone public to-zone private policy P2 then permit
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the device to pass H.323 ALG traffic to a gatekeeper in the internal zone:

1. Configure two address books.

```
[edit]
user@host# set security zones security-zone public address-book address ip_phone
2.2.2.5/32
set security zones security-zone private address-book address gateway 2.2.2.5/32
```

2. Configure policy P1 from the internal zone to the external zone.

```
[edit]
user@host# set security policies from-zone private to-zone public policy P1 match
source-address any
user@host# set security policies from-zone private to-zone public policy P1 match
destination-address IP_Phone
user@host# set security policies from-zone private to-zone public policy P1 match
application junos-h323
user@host# set security policies from-zone private to-zone public policy P1 then
permit
```

3. Configure policy P2 from the external zone to the internal zone.

```
[edit]
user@host# set security policies from-zone public to-zone private policy P2 match
source-address any
user@host# set security policies from-zone public to-zone private policy P2 match
destination-address gateway
user@host# set security policies from-zone public to-zone private policy P2 match
application junos-h323
user@host# set security policies from-zone public to-zone private policy P2 then
permit
```

Results From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this show output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show security policies
...
from-zone trust to-zone trust {
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone trust to-zone untrust {
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
```

```

        }
        then {
            permit;
        }
    }
}
from-zone untrust to-zone trust {
    policy default-deny {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            deny;
        }
    }
}
from-zone private to-zone public {
    policy P1 {
        match {
            source-address any;
            destination-address IP_Phone;
            application junos-h323;
        }
        then {
            permit;
        }
    }
}
from-zone public to-zone private {
    policy P2 {
        match {
            source-address any;
            destination-address gateway;
            application junos-h323;
        }
        then {
            permit;
        }
    }
}
...

```

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying H.323 ALG Configurations on page 187

Verifying H.323 ALG Configurations

Purpose Display information about active calls.



NOTE: H.323 counters for calls and active calls in the output to this show security command do not apply to the proprietary Avaya implementation of H.323. This is because Q.931 setup and connect messages are exchanged right after the phone is powered up and call creation and tear down is done by Facility messages.

Counters for calls and active calls are increased when the resources allocated for calls are increased—that is, messages belonging to the same call and that pass the firewall multiple times increment the counters. This applies when resources for a call need to be allocated multiple times. For example, in a two-zone scenario the setup and connect message pair allocates one call resource, and the active call counter is increased by one. But in a three-zone scenario the setup and connect message pair passes the firewall twice, each time allocating different call resources. In this case, the counter is incremented.

Action From the J-Web interface, select **Monitor>ALGs>H323**. Alternatively, from the CLI, enter the **show security alg h323 counters** command.

Counters for H.245 messages received also will not be accurate in the case of H.245 tunneling. Because H.245 messages are encapsulated in Q.931 packets, the counter for H.245 messages received will remain zero even when there are H.245 messages. The **Other H245** counter will, however, reflect these packet transmissions.

[edit]

```
user@host> show security alg h323 counters
```

H.323 counters summary:

```
Packets received      : 0
Packets dropped       : 0
RAS message received  : 0
Q.931 message received : 0
H.245 message received : 0
Number of calls       : 0
Number of active calls : 0
```

H.323 error counters:

```
Decoding errors       : 0
Message flood dropped  : 0
NAT errors            : 0
Resource manager errors : 0
```

H.323 message counters:

```
RRQ      : 0
RCF      : 0
ARQ      : 0
ACF      : 0
URQ      : 0
UCF      : 0
DRQ      : 0
DCF      : 0
Oth RAS  : 0
Setup    : 0
Alert    : 0
Connect  : 0
CallProd : 0
Info     : 0
RelCmpl  : 0
```

```
Facility : 0
Empty    : 0
OLC      : 0
OLC-ACK  : 0
Oth H245 : 0
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding H.323 ALGs on page 173
 - H.323 ALG Configuration Overview on page 176

Example: Passing H.323 ALG Traffic to a Gatekeeper in the External Zone

This example shows how to set up two policies to allow H.323 traffic to pass between IP phone hosts in the internal zone, and the IP phone at IP address 2.2.2.5/32 (and the gatekeeper) in the external zone.

- Requirements on page 189
- Overview on page 189
- Configuration on page 190
- Verification on page 193

Requirements

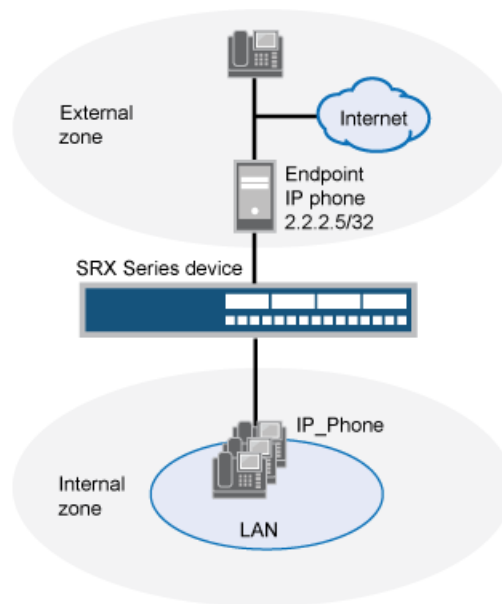
Before you begin:

- Understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.
- Configure security zones. See “Understanding Security Zones” on page 87.

Overview

Because route mode does not require address mapping of any kind, a device configuration for a gatekeeper in the external, or public, zone is usually identical to the configuration for a gatekeeper in an internal, or private, zone. This example shows how to set up two policies to allow H.323 traffic to pass between IP phone hosts in the internal zone, and the IP phone at IP address 2.2.2.5/32 (and the gatekeeper) in the external zone. The device can be in transparent or route mode. See Figure 15 on page 190.

Figure 15: H.323 Gatekeeper in Zone 2



Configuration

CLI Quick Configuration

To quickly configure the device to pass H.323 ALG traffic to a gatekeeper in the external zone, copy the following commands and paste them into the CLI:

```
[edit]
set security zones security-zone external address-book address IP_Phone 2.2.2.5/32
set security zones security-zone internal address-book address gatekeeper 2.2.2.10/32
set security policies from-zone internal to-zone external policy P1 match source-address
any
set security policies from-zone internal to-zone external policy P1 match
destination-address IP_Phone
set security policies from-zone internal to-zone external policy P1 match application
junos-h323
set security policies from-zone internal to-zone external policy P1 then permit
set security policies from-zone internal to-zone external policy P2 match source-address
any
set security policies from-zone internal to-zone external policy P2 match
destination-address gatekeeper
set security policies from-zone internal to-zone external policy P2 match application
junos-h323
set security policies from-zone internal to-zone external policy P2 then permit
set security policies from-zone external to-zone internal policy P3 match source-address
IP_Phone
set security policies from-zone external to-zone internal policy P3 match
destination-address any
set security policies from-zone external to-zone internal policy P3 match application
junos-h323
set security policies from-zone external to-zone internal policy P3 then permit
set security policies from-zone external to-zone internal policy P4 match source-address
gatekeeper
set security policies from-zone external to-zone internal policy P4 match
destination-address any
```

```

set security policies from-zone external to-zone internal policy P4 match application
junos-h323
set security policies from-zone external to-zone internal policy P4 then permit

```

**Step-by-Step
Procedure**

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the device to pass H.323 ALG traffic to a gatekeeper in the external zone:

1. Configure two address books.

```

[edit]
user@host# set security zones security-zone external address-book address
IP_Phone 2.2.2.5/32
user@host# set security zones security-zone internal address-book address
gatekeeper 2.2.2.10/32

```

2. Configure policy P1 from the internal zone to the external zone.

```

[edit]
user@host# set security policies from-zone internal to-zone external policy P1 match
source-address any
user@host# set security policies from-zone internal to-zone external policy P1 match
destination-address IP_Phone
user@host# set security policies from-zone internal to-zone external policy P1 match
application junos-h323
user@host# set security policies from-zone internal to-zone external policy P1 then
permit

```

3. Configure policy P2 to allow traffic between the internal zone and the gatekeeper in the external zone.

```

[edit]
user@host# set security policies from-zone internal to-zone external policy P2 match
source-address any
user@host# set security policies from-zone internal to-zone external policy P2 match
destination-address gatekeeper
user@host# set security policies from-zone internal to-zone external policy P2 match
application junos-h323
user@host# set security policies from-zone internal to-zone external policy P2 then
permit

```

4. Configure policy P3 to allow traffic between phones in the internal zone and the external zone.

```

[edit]
user@host# set security policies from-zone external to-zone internal policy P3 match
source-address IP_Phone
user@host# set security policies from-zone external to-zone internal policy P3 match
destination-address any
user@host# set security policies from-zone external to-zone internal policy P3 match
application junos-h323
user@host# set security policies from-zone external to-zone internal policy P3 then
permit

```

5. Configure policy P4 to allow traffic between phones in the internal zone and the gatekeeper in the external zone.

```
[edit]
user@host# set security policies from-zone external to-zone internal policy P4
match source-address gatekeeper
user@host# set security policies from-zone external to-zone internal policy P4
match destination-address any
user@host# set security policies from-zone external to-zone internal policy P4
match application junos-h323
user@host# set security policies from-zone external to-zone internal policy P4 then
permit
```

Results From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this show output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show security policies
...
from-zone internal to-zone external {
  policy P1 {
    match {
      source-address any;
      destination-address IP_Phone;
      application junos-h323;
    }
    then {
      permit;
    }
  }
  policy P2 {
    match {
      source-address any;
      destination-address gatekeeper;
      application junos-h323;
    }
    then {
      permit;
    }
  }
}
from-zone external to-zone internal {
  policy P3 {
    match {
      source-address IP_Phone;
      destination-address any;
      application junos-h323;
    }
    then {
      permit;
    }
  }
  policy P4 {
```



```

        match {
            source-address gatekeeper;
            destination-address any;
            application junos-h323;
        }
        then {
            permit;
        }
    }
}
...

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying H.323 ALG Configurations on page 193

Verifying H.323 ALG Configurations

Purpose Display information about active calls.



NOTE: H.323 counters for calls and active calls in the output to this **show security** command do not apply to the proprietary Avaya implementation of H.323. This is because Q.931 setup and connect messages are exchanged right after the phone is powered up and call creation and tear down is done by Facility messages.

Counters for calls and active calls are increased when the resources allocated for calls are increased—that is, messages belonging to the same call and that pass the firewall multiple times increment the counters. This applies when resources for a call need to be allocated multiple times. For example, in a two-zone scenario the setup and connect message pair allocates one call resource, and the active call counter is increased by one. But in a three-zone scenario the setup and connect message pair passes the firewall twice, each time allocating different call resources. In this case, the counter is incremented.

Action From the J-Web interface, select **Monitor>ALGs>H323**. Alternatively, from the CLI, enter the **show security alg h323 counters** command.

Counters for H.245 messages received also will not be accurate in the case of H.245 tunneling. Because H.245 messages are encapsulated in Q.931 packets, the counter for H.245 messages received will remain zero even when there are H.245 messages. The **Other H245** counter will, however, reflect these packet transmissions.

```

[edit]
user@host> show security alg h323 counters
H.323 counters summary:
  Packets received      : 0
  Packets dropped       : 0
  RAS message received  : 0
  Q.931 message received : 0

```

```

H.245 message received : 0
Number of calls         : 0
Number of active calls  : 0
H.323 error counters:
Decoding errors         : 0
Message flood dropped   : 0
NAT errors              : 0
Resource manager errors : 0
H.323 message counters:
RRQ                     : 0
RCF                     : 0
ARQ                     : 0
ACF                     : 0
URQ                     : 0
UCF                     : 0
DRQ                     : 0
DCF                     : 0
Oth RAS                 : 0
Setup                   : 0
Alert                   : 0
Connect                 : 0
CallProd                : 0
Info                    : 0
RelCmpl                 : 0
Facility                : 0
Empty                   : 0
OLC                     : 0
OLC-ACK                 : 0
Oth H245                : 0

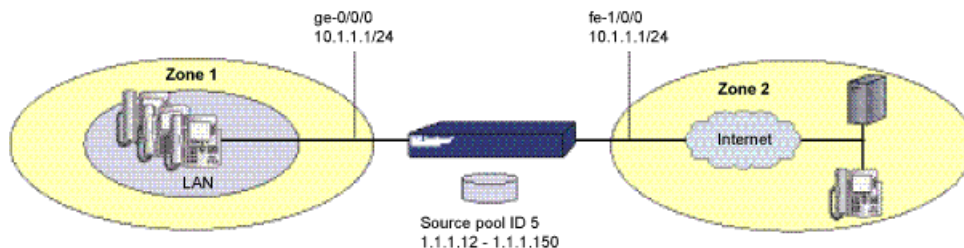
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding H.323 ALGs on page 173
 - H.323 ALG Configuration Overview on page 176

Example: Using NAT and the H.323 ALG to Enable Incoming Calls (CLI)

In this example, you configure the device to accept incoming calls over a NAT boundary. To do this, you can create an interface NAT address pool for dynamically allocating destination addresses. This differs from most configurations, where a source pool provides source addresses only. See Figure 16 on page 194.

Figure 16: Network Address Translation—Incoming Calls



With interface NAT, the source pool uses the same address as an interface IP address. You can use such address entries as destination addresses in policies, together with H.323, SIP, or other VoIP protocols, to support incoming calls.

In the following example, you configure interfaces, a NAT address pool, zones, and security policies for incoming and outgoing traffic:

1. Configure interfaces.

```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces fe-1/0/0 unit 0 family inet address 1.1.1.1/24
```

2. Configure interface NAT.

```
user@host# set security nat source-nat address-persistent
user@host# set security nat interface fe-1/0/0.0 source-nat pool p1 address-range
low 1.1.1.12 high 1.1.1.150
```

3. Configure zones.

```
user@host# set security zones security-zone zone1 interfaces ge-0/0/0.0
user@host# set security zones security-zone zone1 address-book address IP_Phone1
10.1.1.5/32
user@host# set security zones security-zone zone1 address-book address gatekeeper
10.1.1.25/32
user@host# set security zones security-zone zone2 interfaces fe-1/0/0.0
user@host# set security zones security-zone zone2 address-book address IP_Phone2
2.2.2.5/32
user@host# set security zones Global
```

4. Configure policies for outgoing traffic.

```
user@host# set security policy from-zone zone1 to-zone zone2 policy zone1_to_zone2
match source-address IP_Phone1
user@host# set security policy from-zone zone1 to-zone zone2 policy zone1_to_zone2
match source-address gatekeeper
user@host# set security policy from-zone zone1 to-zone zone2 policy zone1_to_zone2
match destination-address IP_Phone2
user@host# set security policy from-zone zone1 to-zone zone2 policy zone1_to_zone2
match application junos-h323
user@host# set security policy from-zone zone1 to-zone zone2 policy zone1_to_zone2
then permit source-nat pool p1
```

5. Configure policies for incoming traffic.

```
user@host# set security policy from-zone zone2 to-zone Global policy zone2_to_Global
match source-address IP_Phone2
user@host# set security policy from-zone zone2 to-zone Global policy zone2_to_Global
match destination-address incoming_nat_p1
user@host# set security policy from-zone zone2 to-zone Global policy zone2_to_Global
match application junos-h323
user@host# set security policy from-zone zone2 to-zone Global policy zone2_to_Global
then permit
```

6. If you are finished configuring the device, commit the configuration.

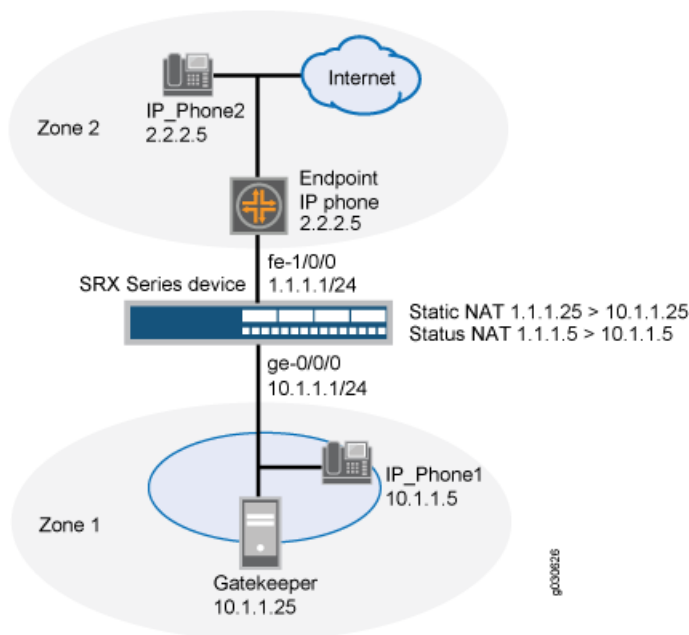
- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding H.323 ALGs on page 173
 - H.323 ALG Configuration Overview on page 176

Example: Using NAT and the H.323 ALG to Enable Outgoing Calls (CLI)

In this example, the devices in the external zone include the endpoint host (10.1.1.5) and the gatekeeper (10.1.1.25). IP_Phone2 (2.2.2.5) is in Zone 2. You configure the device to allow traffic between the endpoint host IP_Phone1 and the gatekeeper in the external zone and the endpoint host IP_Phone2 in the internal zone.

When the Juniper Networks device uses NAT, a gatekeeper or endpoint device in the external zone has a private address, and when it is in the internal zone, it has a public address. See Figure 17 on page 196.

Figure 17: Network Address Translation—Outgoing Calls



1. Configure interfaces.

```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces fe-1/0/0 unit 0 family inet address 1.1.1.1/24
user@host# set security zones security-zone zone1 interfaces ge-0/0/0.0
user@host# set security zones security-zone zone2 interfaces fe-1/0/0.0
```

2. Configure zones.

```
user@host# set security zones security-zone zone1 interfaces ge-0/0/0.0
user@host# set security zones security-zone zone1 address-book address IP_Phone1
10.1.1.5/32
user@host# set security zones security-zone zone1 address-book address gatekeeper
10.1.1.25/32
user@host# set security zones security-zone zone2 interfaces fe-1/0/0.0
user@host# set security zones security-zone zone2 address-book address IP_Phone2
2.2.2.5/32
user@host# set security zones Global
```

3. Configure interface NAT.

```

user@host# set security nat interface fe-1/0/0.0 static-nat 1.1.1.5/32 host 10.1.1.5/32
user@host# set security nat interface fe-1/0/0.0 static-nat 1.1.1.25/32 host 10.1.1.25/32

```

4. Configure policies.

```

user@host# set security policy from-zone zone1 to-zone zone2 policy zone1_to_zone2
match source-address IP_Phone1
user@host# set security policy from-zone zone1 to-zone zone2 policy zone1_to_zone2
match source-address gatekeeper
user@host# set security policy from-zone zone1 to-zone zone2 policy zone1_to_zone2
match destination-address IP_Phone2
user@host# set security policy from-zone zone1 to-zone zone2 policy zone1_to_zone2
match application junos-h323
user@host# set security policy from-zone zone1 to-zone zone2 policy zone1_to_zone2
then permit
user@host# set security policy from-zone zone2 to-zone Global policy zone2_to_Global
match source-address IP_Phone2
user@host# set security policy from-zone zone2 to-zone Global policy zone2_to_Global
match destination-address static_nat_1.1.1.5_32
user@host# set security policy from-zone zone2 to-zone Global policy zone2_to_Global
match destination-address static_nat_1.1.1.25_32
user@host# set security policy from-zone zone2 to-zone Global policy zone2_to_Global
match application junos-h323
user@host# set security policy from-zone zone2 to-zone Global policy zone2_to_Global
then permit

```

5. If you are finished configuring the device, commit the configuration.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding H.323 ALGs on page 173
 - H.323 ALG Configuration Overview on page 176

CHAPTER 11

ALG for IKE and ESP

- Understanding ALG for IKE and ESP on page 199
- Understanding ALG for IKE and ESP Operation on page 200
- Example: Configuring the IKE and ESP ALG (CLI) on page 200
- Example: Enabling IKE and ESP ALG and Setting Timeouts (CLI) on page 204

Understanding ALG for IKE and ESP

An SRX Series or J Series device can be used solely as a Network Address Translation (NAT) device when placed between VPN clients on the private side of the NAT gateway and the virtual private network (VPN) gateways on the public side.

Internet Key Exchange (IKE) and Encapsulating Security Payload (ESP) traffic is exchanged between the clients and the server. However, if the clients do not support NAT-Traversal (NAT-T) and if the device assigns the same NAT-generated IP address to two or more clients, the device will be unable to distinguish and route return traffic properly.



NOTE: If the user wants to support both NAT-T-capable and non-NAT-T-capable clients, then some additional configurations are required. If there are NAT-T capable clients, the user must enable the source NAT address persistence.

ALG for IKE and ESP monitors IKE traffic between the client and the server and permits only one IKE Phase 2 message exchange between any given client/server pair, not just one exchange between any client and any server.

ALG for IKE and ESP traffic has been created and NAT has been enhanced to implement the following:

- To enable the SRX Series and J Series devices to pass IKE and ESP traffic with a source NAT pool
- To allow the device to be configured to return the same NAT-generated IP address for the same IP address without NAT ("address-persistent NAT"). As a result, the device is able to associate a client's outgoing IKE traffic with its return traffic from the server, especially when the IKE session times out and needs to be reestablished.

- The resulting ESP traffic between the client and the server is also allowed, especially in the direction from the server to the client.
- The return ESP traffic matches the following:
 - The server IP address as source IP
 - The client IP address as destination IP

Understanding ALG for IKE and ESP Operation

The proposed ALG for IKE and ESP traffic will have the following behavior:

- The ALG for IKE and ESP monitors IKE traffic between the client and the server, and permits only one IKE phase 2 message exchange between the client and the server at any given time.
- When a phase 2 message is seen:
 - If no phase 2 exchange between the client and server is already taking place, the IKE ALG will open gates for the relevant ESP traffic in the client to server and server to client directions.
 - If the gates cannot be successfully opened, or if there is already a phase 2 exchange taking place, the phase 2 message will be dropped.
- When ESP traffic hits those gates, sessions will be created to capture subsequent ESP traffic, and perform the proper NATing (source IP address translation for client ->server traffic, and destination IP address translation for server->client traffic).
- If no traffic hits either or both of the gates, the gate(s) will naturally time out.
- Once the gates are collapsed or timed out, another IKE phase 2 exchange will be permitted.
- IKE NAT-T traffic on floating port 4500 will not be processed in IKE ALG. To support mixture of NAT-T-capable and non-capable clients, users is required to enable source NAT address persistent.

- Related Topics**
- [ALG Overview on page 169](#)
 - [NAT Overview on page 927](#)
 - [Understanding ALG for IKE and ESP on page 199](#)
 - [Example: Configuring the IKE and ESP ALG \(CLI\) on page 200](#)
 - [Example: Enabling IKE and ESP ALG and Setting Timeouts \(CLI\) on page 204](#)

Example: Configuring the IKE and ESP ALG (CLI)

In this example, you configure the IKE/ESP ALG on the device.

To configure the IKE/ESP ALG:

1. Configure a source NAT pool..

```
[edit]
user@host# set security nat source pool p1 address 10.10.10.1/32 to 10.10.10.10/32
user@host# set security nat source rule-set rs1 from zone green
user@host# set security nat source rule-set rs1 to zone red
user@host# set security nat source rule-set rs1 rule r1 match source-address 1.1.1.0/24
user@host# set security nat source rule-set rs1 rule r1 match destination-address
2.2.2.0/24
user@host# set security nat source rule-set rs1 rule r1 then source-nat pool p1
```

Proxy ARP also needs to be configured for all IP addresses in the source NAT pool.

2. Confirm your configuration by entering the **show security nat** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool p1 {
    address {
      10.10.10.1/32 to 10.10.10.10/32;
    }
  }
  address-persistent;
  rule-set rs1 {
    from zone green;
    to zone red;
    rule r1 {
      match {
        source-address 1.1.1.0/24;
        destination-address 2.2.2.0/24;
      }
      then {
        source-nat {
          pool {
            p1;
          }
        }
      }
    }
  }
}
```

3. Commit the configuration if you are done configuring the device.

```
[edit]
user@host# edit security nat
user@host# commit
```

4. Configure a custom application.

```
[edit]
user@host# set applications application custom-ike-alg source-port 500
destination-port 500 protocol udp application-protocol ike-esp-nat
```

5. Confirm your configuration by entering the **show applications** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show applications
application custom-ike-alg {
  application-protocol ike-esp-nat;
  protocol udp;
  source-port 500;
  destination-port 500;
}
```

6. Commit the configuration if you are done configuring the device.

```
[edit]
user@host# edit security applications
user@host# commit
```

7. Associate the custom application configured using a policy.

```
[edit]
user@host# set security zones security-zone green address-book address sa1 1.1.1.0/24
user@host# set security zones security-zone red address-book address da1 2.2.2.0/24
user@host# set security policies from-zone green to-zone red policy pol1 match
  source-address sa1
user@host# set security policies from-zone green to-zone red policy pol1 match
  destination-address da1
user@host# set security policies from-zone green to-zone red policy pol1 match
  application custom-ike-alg
user@host# set security policies from-zone green to-zone red policy pol1 then permit
```

8. Confirm your configuration by entering the **show security zones** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security zones
security-zone Trust {
  host-inbound-traffic {
    system-services {
      all;
    }
    protocols {
      all;
    }
  }
  interfaces {
    ge-0/0/1.0;
  }
}
security-zone green {
  address-book {
    address sa1 1.1.1.0/24;
  }
}
security-zone red {
  address-book {
```

```

        address da1 2.2.2.0/24;
    }
}

```

9. Commit the configuration if you are done configuring the device.

```

[edit]
user@host# edit security zones
user@host# commit

```

If users want to support both NAT-T-capable and non-capable clients, they need some additional configurations.

1. Globally enable persistent source NAT translation (so that once a particular source NAT is associated with a given IP address, subsequent source NAT translations use the same IP address).

```

[edit]
user@host# set security nat source address-persistent

```

2. Confirm your configuration by entering the **show applications** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security nat
source {
    address-persistent;
}

```

3. Commit the configuration if you are done configuring the device.

```

[edit]
user@host# edit security nat
user@host# commit

```

4. Configure the IKE NAT-T application.

```

[edit]
user@host# set applications application custom-ike-natt protocol udp source-port
4500 destination-port 4500

```

5. Confirm your configuration by entering the **show applications** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show applications
application custom-ike-natt {
    protocol udp;
    source-port 4500;
    destination-port 4500;
}

```

6. Commit the configuration if you are done configuring the device.

```

[edit]
user@host# edit security applications
user@host# commit

```

7. Associate the NAT-T application using a policy.

```
[edit]
user@host# set security policies from-zone green to-zone red policy pol1 match
source-address sa1
user@host# set security policies from-zone green to-zone red policy pol1 match
destination-address da1
user@host# set security policies from-zone green to-zone red policy pol1 match
application custom-ike-natt
user@host# set security policies from-zone green to-zone red policy pol1 then permit
```

8. Confirm your configuration by entering the **show security policies** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
from-zone green to-zone red {
  policy pol1 {
    match {
      source-address sa1;
      destination-address da1;
      application [ custom-ike-alg custom-ike-natt ];
    }
    then {
      permit;
    }
  }
}
default-policy {
  permit-all;
}
```

9. Commit the configuration if you are done configuring the device.

```
[edit]
user@host# edit security policies
user@host# commit
```

- Related Topics**
- ALG Overview on page 169
 - NAT Overview on page 927
 - Understanding ALG for IKE and ESP on page 199
 - Understanding ALG for IKE and ESP Operation on page 200
 - Example: Enabling IKE and ESP ALG and Setting Timeouts (CLI) on page 204

Example: Enabling IKE and ESP ALG and Setting Timeouts (CLI)

In the following example, you enable the IKE/ESP ALG and set timeouts.

1. To enable the IKE ESP ALG, set this CLI.

The IKE ESP ALG will handle all traffic specified in any policy to which the ALG is attached. Additionally, if this CLI is present, the current default IPsec pass-through

behavior will be disabled for all IPsec pass-through traffic, regardless of policy. If this CLI is NOT set, IKE ESP ALG will be disabled, and IPsec pass-through traffic will be handled following the default IPsec pass-through behavior.

```
[edit]
user@host# edit security alg ike-esp-nat
user@host# set enable
```

2. The **state-timeout** sets the timeout of ALG state information. ALG state information will be aged out using this timeout value. The timeout range is 180 through 86400 seconds. The default timeout is 14400 seconds.

```
[edit]
user@host# edit security alg ike-esp-nat
user@host# set state-timeout 360
```

3. The **esp-gate-timeout** sets the timeout of the ESP gates created after a phase 2 exchange has completed. The timeout range is 2 through 30 seconds. The default timeout is 5 seconds.

```
[edit]
user@host# edit security alg ike-esp-nat
user@host# set esp-gate-timeout 20
```

4. The **esp-session-timeout** sets the idle timeout of the ESP sessions created from the IPsec gates; if no traffic hits the session, it will be aged out after this period of time. The timeout range is 60 through 2400 seconds. The default timeout is 1800 seconds.

```
[edit]
user@host# edit security alg ike-esp-nat
user@host# set esp-session-timeout 2400
```

5. Confirm your configuration by entering the **show security alg** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security alg
ike-esp-nat {
    enable;
    state-timeout 360;
    esp-gate-timeout 20;
    esp-session-timeout 2400;
}
```

6. Commit the configuration if you are done configuring the device.

```
[edit]
user@host# edit security alg
user@host# commit
```

Related Topics

- ALG Overview on page 169
- NAT Overview on page 927
- Understanding ALG for IKE and ESP on page 199
- Understanding ALG for IKE and ESP Operation on page 200

- Example: Configuring the IKE and ESP ALG (CLI) on page 200

CHAPTER 12

SIP ALGs

- Understanding SIP ALGs on page 207
- Understanding SIP ALG Request Methods on page 212
- SIP ALG Configuration Overview on page 213
- SIP ALG Call Duration and Timeouts on page 213
- SIP ALG DoS Attack Protection on page 216
- SIP ALG Unknown Message Types on page 217
- SIP ALG Hold Resources on page 219
- SIP ALGs and NAT on page 220
- Verifying SIP ALG Configurations on page 241

Understanding SIP ALGs

Session Initiation Protocol (SIP) is an Internet Engineering Task Force (IETF)-standard protocol for initiating, modifying, and terminating multimedia sessions over the Internet. Such sessions might include conferencing, telephony, or multimedia, with features such as instant messaging and application-level mobility in network environments.

Junos OS supports SIP as a service and screens SIP traffic, allowing and denying it based on a policy that you configure. SIP is a predefined service in Junos OS and uses port 5060 as the destination port.

SIP's primary function is to distribute session-description information and, during the session, to negotiate and modify the parameters of the session. SIP is also used to terminate a multimedia session.

Session-description information is included in INVITE and ACK messages and indicates the multimedia type of the session; for example, whether it is voice or video. Although SIP can use different description protocols to describe the session, the Juniper Networks SIP Application Layer Gateway (ALG) supports only the Session Description Protocol (SDP).

SDP provides information that a system can use to join a multimedia session. SDP might include information such as IP addresses, port numbers, times, and dates. Note that the IP address and port number in the SDP header (the c= and m= fields, respectively) are the address and port where the client wants to receive the media streams and not the

IP address and port number from which the SIP request originates (although they can be the same).

SIP messages consist of requests from a client to a server and responses to the requests from a server to a client with the purpose of establishing a session (or a call). A User Agent (UA) is an application that runs at the endpoints of the call and consists of two parts:

- User Agent Client (UAC), which sends SIP requests on behalf of the user
- User Agent Server (UAS), which listens to the responses and notifies the user when they arrive

Examples of UAs are SIP proxy servers and phones.

This topic contains the following sections:

- SIP ALG Operation on page 208
- SDP Session Descriptions on page 209
- Pinhole Creation on page 210

SIP ALG Operation

There are two types of SIP traffic, the signaling and the media stream. SIP signaling traffic consists of request and response messages between client and server and uses transport protocols such as UDP or TCP. The media stream carries the data (audio data, for example) and uses Application Layer protocols such as Real-Time Transport Protocol (RTP) over UDP.

Junos OS supports SIP signaling messages on port 5060. You can simply create a policy that permits SIP service, and the software filters SIP signaling traffic like any other type of traffic, permitting or denying it. The media stream, however, uses dynamically assigned port numbers that can change several times during the course of a call. Without fixed ports, it is impossible to create a static policy to control media traffic. In this case, the device invokes the SIP ALG. The SIP ALG reads SIP messages and their SDP content and extracts the port-number information it needs to dynamically open pinholes to let the media stream traverse the device.



NOTE: We refer to a pinhole as the limited opening of a port to allow exclusive traffic.

The SIP ALG monitors SIP transactions and dynamically creates and manages pinholes based on the information it extracts from these transactions. The Juniper Networks SIP ALG supports all SIP methods and responses. You can allow SIP transactions to traverse the Juniper Networks firewall by creating a static policy that permits SIP service. This policy enables the device to intercept SIP traffic and do one of the following actions: permit or deny the traffic or enable the SIP ALG to open pinholes to pass the media stream. The SIP ALG needs to open pinholes only for the SIP requests and responses that contain media information (SDP). For SIP messages that do not contain SDP, the device simply lets them through.

The SIP ALG intercepts SIP messages that contain SDP and, using a parser, extracts the information it requires to create pinholes. The SIP ALG examines the SDP portion of the packet, and a parser extracts information such as IP addresses and port numbers, which the SIP ALG records in a pinhole table. The SIP ALG uses the IP addresses and port numbers recorded in the pinhole table to open pinholes and allow media streams to traverse the device.



NOTE: Junos OS does not support encrypted SDP. If your device receives a SIP message in which SDP is encrypted, the SIP ALG permits it through the firewall but generates a log message informing the user that it cannot process the packet. If SDP is encrypted, the SIP ALG cannot extract the information it needs from SDP to open pinholes. As a result, the media content that SDP describes cannot traverse the device.

We support NEC standards and when we implement SIP-NEC ALG, NEC engineers come to our site and setup the environment.

The SIP NEC support includes:

- SIP NEC server which is CC100 series
- SIP NEC hard-phone/soft-phone series:
 - SIP NEC hard-phone: NEC NETerm50 SIP Phone
 - SIP NEC soft-phone: NEC DetermSP30 Softphone

SDP Session Descriptions

An SDP session description is text-based and consists of a set of lines. It can contain session-level and media-level information. The session-level information applies to the whole session, while the media-level information applies to a particular media stream. An SDP session description always contains session-level information, which appears at the beginning of the description, and might contain media-level information, which comes after.



NOTE: In the SDP session description, the media-level information begins with the `m=` field.

Of the many fields in the SDP description, two are particularly useful to the SIP ALG because they contain Transport Layer information.

- `c=` for connection information

This field can appear at the session or media level. It displays in this format:

`c=<network-type><address-type><connection-address>`

Currently, Junos OS supports only “IN” (for Internet) as the network type, “IP4” as the address type, and a unicast IP address or domain name as the destination (connection) IP address.

If the destination IP address is a unicast IP address, the SIP ALG creates pinholes using the IP address and port numbers specified in the media description field `m=`.

- `m=` for media announcement

This field appears at the media level and contains the description of the media. It displays in this format:

`m=<media><port><transport><fmt list>`

Currently, the Junos OS supports only “audio” as the media and “RTP” as the Application Layer transport protocol. The port number indicates the destination (not the origin) of the media stream. The format list (fmt list) provides information on the Application Layer protocol that the media uses.

The software opens ports only for RTP and Real-Time Control Protocol (RTCP). Every RTP session has a corresponding RTCP session. Therefore, whenever a media stream uses RTP, the SIP ALG must reserve ports (create pinholes) for both RTP and RTCP traffic. By default, the port number for RTCP is one higher than the RTP port number.

Pinhole Creation

Both pinholes for the RTP and RTCP traffic share the same destination IP address. The IP address comes from the `c=` field in the SDP session description. Because the `c=` field can appear in either the session-level or media-level portion of the SDP session description, the parser determines the IP address based on the following rules (in accordance with SDP conventions):

- First, the SIP ALG parser verifies if there is a `c=` field containing an IP address in the media level. If there is one, the parser extracts that IP address, and the SIP ALG uses it to create a pinhole for the media.
- If there is no `c=` field in the media level, the SIP ALG parser extracts the IP address from the `c=` field in the session level, and the SIP ALG uses it to create a pinhole for the media. If the session description does not contain a `c=` field in either level, this indicates an error in the protocol stack, and the device drops the packet and logs the event.

The SIP ALG needs the following information to create a pinhole. This information comes from the SDP session description and parameters on the device:

- Protocol—UDP.
- Source IP—Unknown.
- Source port—Unknown.
- Destination IP—The parser extracts the destination IP address from the `c=` field in the media or session level.
- Destination port—The parser extracts the destination port number for RTP from the `m=` field in the media level and calculates the destination port number for RTCP using the following formula:

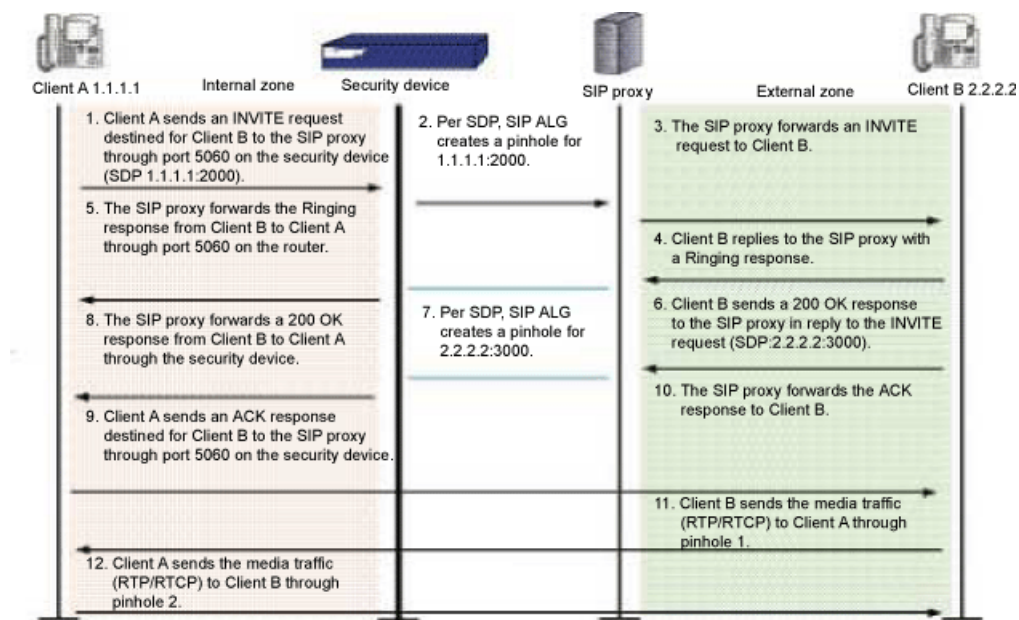
RTP port number + one

- **Lifetime**—This value indicates the length of time (in seconds) during which a pinhole is open to allow a packet through. A packet must go through the pinhole before the lifetime expires. When the lifetime expires, the SIP ALG removes the pinhole.

When a packet goes through the pinhole within the lifetime period, immediately afterwards the SIP ALG removes the pinhole for the direction from which the packet came.

Figure 18 on page 211 describes a call setup between two SIP clients and how the SIP ALG creates pinholes to allow RTP and RTCP traffic. The illustration assumes that the device has a policy that permits SIP, thus opening port 5060 for SIP signaling messages.

Figure 18: SIP ALG Call Setup



NOTE: The SIP ALG does not create pinholes for RTP and RTCP traffic when the destination IP address is 0.0.0.0, which indicates that the session is on hold. To put a session on hold during a telephone communication, for example, user A sends user B a SIP message in which the destination IP address is 0.0.0.0. Doing so indicates to user B that it should not send any media until further notice. If user B sends media anyway, the device drops the packets.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - ALG Overview on page 169
 - Understanding SIP ALG Request Methods on page 212
 - Understanding SIP ALGs and NAT on page 221
 - SIP ALG Configuration Overview on page 213

Understanding SIP ALG Request Methods

The Session Initiation Protocol (SIP) transaction model includes a number of request and response messages, each of which contains a *method* field that denotes the purpose of the message.

Junos OS supports the following method types and response codes:

- **INVITE**—A user sends an INVITE request to invite another user to participate in a session. The body of an INVITE request can contain the description of the session.
- **ACK**—The user from whom the INVITE originated sends an ACK request to confirm reception of the final response to the INVITE request. If the original INVITE request did not contain the session description, the ACK request must include it.
- **OPTIONS**—The User Agent (UA) obtains information about the capabilities of the SIP proxy. A server responds with information about what methods, session description protocols, and message encoding it supports.
- **BYE**—A user sends a BYE request to abandon a session. A BYE request from either user automatically terminates the session.
- **CANCEL**—A user sends a CANCEL request to cancel a pending INVITE request. A CANCEL request has no effect if the SIP server processing the INVITE had sent a final response for the INVITE before it received the CANCEL.
- **REGISTER**—A user sends a REGISTER request to a SIP registrar server to inform it of the current location of the user. A SIP registrar server records all the information it receives in REGISTER requests and makes this information available to any SIP server attempting to locate a user.
- **Info**—Used to communicate mid-session signaling information along the signaling path for the call.
- **Subscribe**—Used to request current state and state updates from a remote node.
- **Notify**—Sent to inform subscribers of changes in state to which the subscriber has a subscription.
- **Refer**—Used to refer the recipient (identified by the Request-URI) to a third party by the contact information provided in the request.

For example, if user A in a private network refers user B, in a public network, to user C, who is also in the private network, the SIP Application Layer Gateway (ALG) allocates a new IP address and port number for user C so that user C can be contacted by user B. If user C is registered with a registrar, however, its port mapping is stored in the ALG Network Address Translation (NAT) table and is reused to perform the translation.

- **Update**—Used to open pinhole for new or updated SDP information. The Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified.
- **1xx, 202, 2xx, 3xx, 4xx, 5xx, 6xx Response Codes**—Used to indicate the status of a transaction. Header fields are modified.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - ALG Overview on page 169
 - Understanding SIP ALGs on page 207

SIP ALG Configuration Overview

The Session Initiation Protocol Application Layer Gateway (SIP ALG) is enabled by default on the device—no action is required to enable it. However, you might choose to fine-tune SIP ALG operations by using the following instructions:

1. Control SIP call activity. For instructions, see “Example: Setting SIP ALG Call Duration and Timeouts” on page 214.
2. Protect the SIP proxy server from denial-of-service (DoS) flood attacks. For instructions, see “Example: Configuring SIP ALG DoS Attack Protection” on page 216.
3. Enable unknown messages to pass when the session is in Network Address Translation (NAT) mode and route mode. For instructions, see “Example: Allowing Unknown SIP ALG Message Types” on page 218.
4. Accommodate proprietary SIP call flows. For instructions, see:
 - Retaining SIP ALG Hold Resources (J-Web Procedure) on page 220
 - Retaining SIP ALG Hold Resources (CLI Procedure) on page 220

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALGs on page 207
 - Understanding SIP ALGs and NAT on page 221
 - Verifying SIP ALG Configurations on page 241

SIP ALG Call Duration and Timeouts

- Understanding SIP ALG Call Duration and Timeouts on page 213
- Example: Setting SIP ALG Call Duration and Timeouts on page 214

Understanding SIP ALG Call Duration and Timeouts

The call duration and timeout features give you control over Session Initiation Protocol (SIP) call activity and help you to manage network resources.

Typically a call ends when one of the clients sends a BYE or CANCEL request. The SIP Application Layer Gateway (ALG) intercepts the BYE or CANCEL request and removes all media sessions for that call. There could be reasons or problems preventing clients in a call from sending BYE or CANCEL requests, for example, a power failure. In this case, the call might go on indefinitely, consuming resources on the device.

A call can have one or more voice channels. Each voice channel has two sessions (or two media streams), one for Real-Time Transport Protocol (RTP) traffic and one for Real-Time Control Protocol (RTCP) signaling. When managing the sessions, the device considers the sessions in each voice channel as one group. Timeouts and call duration settings apply to a group as opposed to each session.

The following parameters govern SIP call activity:

- **inactive-media-timeout**—This parameter indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the temporary openings (pinholes) in the firewall the SIP ALG opened for media are closed. The default setting is 120 seconds, and the range is 10 through 2550 seconds. Note that upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated.
- **maximum-call-duration**—This parameter sets the absolute maximum length of a call. When a call exceeds this parameter setting, the SIP ALG tears down the call and releases the media sessions. The default setting is 43200 seconds, and the range is 180 through 432000 seconds.
- **t1-interval**—This parameter specifies the roundtrip time estimate, in seconds, of a transaction between endpoints. The default is 500 milliseconds. Because many SIP timers scale with the t1-interval (as described in RFC 3261), when you change the value of the t1-interval timer, those SIP timers also are adjusted.
- **t4-interval**—This parameter specifies the maximum time a message remains in the network. The default is 5 seconds and the range is 5 through 10 seconds. Because many SIP timers scale with the t4-interval (as described in RFC 3261), when you change the value of the t4-interval timer, those SIP timers also are adjusted.
- **c-timeout**—This parameter specifies the INVITE transaction timeout at the proxy, in minutes; the default is 3. Because the SIP ALG is in the middle, instead of using the INVITE transaction timer value B (which is $(64 * T1) = 32$ seconds), the SIP ALG gets its timer value from the proxy.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding SIP ALGs on page 207
- SIP ALG Configuration Overview on page 213
- Example: Setting SIP ALG Call Duration and Timeouts on page 214

Example: Setting SIP ALG Call Duration and Timeouts

This example shows how to set the call duration and the media inactivity timeout.

Requirements

Before you begin, review the call duration and timeout features used to control SIP call activity. See “Understanding SIP ALG Call Duration and Timeouts” on page 213.

Overview

The call duration and inactivity media timeout features help you to conserve network resources and maximize throughput.

The **maximum-call-duration** parameter sets the maximum allowable length of time a call can be active. When the duration is exceeded, the SIP ALG tears down the call and releases the media sessions. This setting also frees up bandwidth in cases where calls fail to properly terminate.

The **inactive-media-timeout** parameter indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the SIP ALG temporary openings (pinholes) for media in the firewall are closed. The default setting is 120 seconds, and the range is 10 through 2550 seconds. Upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated.

In this example, the call duration is set to 180000 seconds and the media inactivity timeout is set to 90 seconds.

Configuration

J-Web Quick Configuration

To set the SIP ALG call duration and the media inactivity timeout:

1. Select **Configure>Security>ALG**.
2. Select the **SIP** tab.
3. In the Maximum call duration field, type **3000**.
4. In the Inactive media timeout field, enter **90**.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To set the SIP ALG call duration and the media inactivity timeout:

1. Configure the SIP ALG call duration.

```
[edit]
user@host# set security alg sip maximum-call-duration 3000
```
2. Configure the SIP ALG inactivity media timeout.

```
[edit]
user@host# set security alg sip inactive-media-timeout 90
```
3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg sip** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - SIP ALG Configuration Overview on page 213
 - Verifying SIP ALG Configurations on page 241

SIP ALG DoS Attack Protection

- Understanding SIP ALG DoS Attack Protection on page 216
- Example: Configuring SIP ALG DoS Attack Protection on page 216

Understanding SIP ALG DoS Attack Protection

The ability of the Session Initiation Protocol (SIP) proxy server to process calls can be impacted by repeat SIP INVITE requests—requests that it initially denied. The denial-of-service (DoS) protection feature enables you to configure the device to monitor INVITE requests and proxy server replies to them. If a reply contains a 3xx, 4xx, or 5xx response code (see “Classes of SIP Responses” on page 228), the ALG stores the source IP address of the request and the IP address of the proxy server in a table. Subsequently, the device checks all INVITE requests against this table and, for a configurable number of seconds (the default is 3), discards any packets that match entries in the table. You can configure the device to monitor and deny repeat INVITE requests to all proxy servers, or you can protect a specific proxy server by specifying the destination IP address. SIP attack protection is configured globally.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALGs on page 207
 - SIP ALG Configuration Overview on page 213
 - Example: Configuring SIP ALG DoS Attack Protection on page 216

Example: Configuring SIP ALG DoS Attack Protection

This example shows how to configure the DoS attack protection feature.

Requirements

Before you begin, review the DoS attack protection feature used to control SIP call activity. See “Understanding SIP ALG DoS Attack Protection” on page 216.

Overview

The ability of the SIP proxy server to process calls can be impacted by repeat SIP INVITE requests—requests that the server initially denied. The DoS protection feature enables you to configure the device to monitor INVITE requests and proxy server replies to them.

In this example, the device is configured to protect a single SIP proxy server (1.1.1.3) from repeat INVITE requests to which it has already been denied service. Packets are dropped

for a period of 5 seconds, after which the device resumes forwarding INVITE requests from those sources.

Configuration

J-Web Quick Configuration

To configure SIP ALG attack protection:

1. Select **Configure>Security>ALG**.
2. Select the **SIP** tab.
3. In the Enable attack protection area, click the **Selected servers** option.
4. In the Destination IP box, enter **1.1.1.3** and click **Add**.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure SIP ALG attack protection:

1. Configure the device to protect a single SIP proxy server.

```
[edit]
user@host# set security alg sip application-screen protect deny destination-ip 1.1.1.3
```
2. Configure the device for the deny timeout period.

```
[edit]
user@host# set security alg sip application-screen protect deny timeout 5
```
3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg sip** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- SIP ALG Configuration Overview on page 213
- Verifying SIP ALG Configurations on page 241

SIP ALG Unknown Message Types

- Understanding SIP ALG Unknown Message Types on page 217
- Example: Allowing Unknown SIP ALG Message Types on page 218

Understanding SIP ALG Unknown Message Types

This feature enables you to specify how unidentified Session Initiation Protocol (SIP) messages are handled by the device. The default is to drop unknown (unsupported) messages.

We do not recommend permitting unknown messages because they can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment. Permitting unknown SIP messages can help you get your network operational so you can later analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped. The unknown SIP message type feature enables you to configure the device to accept SIP traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.



NOTE: This option applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALGs on page 207
 - SIP ALG Configuration Overview on page 213
 - Example: Allowing Unknown SIP ALG Message Types on page 218

Example: Allowing Unknown SIP ALG Message Types

This example shows how to allow unknown message types.

Requirements

Before you begin, review how unidentified SIP messages are handled by the device. See “Understanding SIP ALG Unknown Message Types” on page 217.

Overview

In this example, you configure the device to allow unknown message types in SIP traffic in both NAT mode and route mode. The default is to drop unknown (unsupported) messages.

Configuration

J-Web Quick Configuration

To allow unknown SIP ALG message types:

1. Select **Configure>Security>ALG**.
2. Select the **SIP** tab.
3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To allow unknown SIP ALG message types:

1. Configure the device to allow unknown message types in SIP traffic.

```
[edit]
user@host# set security alg sip application-screen unknown-message
permit-nat-applied permit-routed
```
2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg sip** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - SIP ALG Configuration Overview on page 213
 - Verifying SIP ALG Configurations on page 241

SIP ALG Hold Resources

- Understanding SIP ALG Hold Resources on page 219
- Retaining SIP ALG Hold Resources (J-Web Procedure) on page 220
- Retaining SIP ALG Hold Resources (CLI Procedure) on page 220

Understanding SIP ALG Hold Resources

When a user puts a call on hold, the Session Initiation Protocol Application Layer Gateway (SIP ALG) releases Session Description Protocol (SDP) media resources, such as pinholes and translation contexts. When the user resumes the call, an INVITE request message negotiates a new SDP offer and answer and the SIP ALG reallocates resources for the media stream. This can result in new translated IP address and port numbers for the media description even when the media description is the same as the previous description. This is compliant with *RFC 3264 An Offer/Answer Model with the Session Description Protocol (SDP)*.

Some proprietary SIP implementations have designed call flows so that the User Agent (UA) module ignores the new SDP INVITE offer and continues to use the SDP offer of the previous negotiation. To accommodate this functionality, you must configure the device to retain SDP media resources when a call is put on hold for reuse when the call is resumed.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALGs on page 207
 - SIP ALG Configuration Overview on page 213
 - Retaining SIP ALG Hold Resources (J-Web Procedure) on page 220

- Retaining SIP ALG Hold Resources (CLI Procedure) on page 220

Retaining SIP ALG Hold Resources (J-Web Procedure)

To accommodate proprietary SIP call flows:

1. Select **Configure>Security>ALG**.
2. Select the **SIP** tab.
3. Select the **Enable retail hold resource** check box.
4. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALG Hold Resources on page 219
 - SIP ALG Configuration Overview on page 213
 - Retaining SIP ALG Hold Resources (CLI Procedure) on page 220
 - Verifying SIP ALG Configurations on page 241

Retaining SIP ALG Hold Resources (CLI Procedure)

To accommodate proprietary SIP call flows:

```
user@host# set security alg sip retain-hold-resource
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALG Hold Resources on page 219
 - SIP ALG Configuration Overview on page 213
 - Retaining SIP ALG Hold Resources (J-Web Procedure) on page 220
 - Verifying SIP ALG Configurations on page 241

SIP ALGs and NAT

- Understanding SIP ALGs and NAT on page 221
- Understanding Incoming SIP ALG Call Support Using the SIP Registrar and NAT on page 229
- Example: Configuring Interface Source NAT for Incoming SIP Calls (CLI) on page 230
- Example: Configuring a Source NAT Pool for Incoming SIP Calls (CLI) on page 232
- Example: Configuring Static NAT for Incoming SIP Calls (CLI) on page 234
- Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone (CLI) on page 235

- Example: Configuring the SIP Proxy and NAT in the Public Zone (CLI) on page 237
- Example: Configuring a Three-Zone SIP ALG and NAT Scenario (CLI) on page 238

Understanding SIP ALGs and NAT

The Network Address Translation (NAT) protocol enables multiple hosts in a private subnet to share a single public IP address to access the Internet. For outgoing traffic, NAT replaces the private IP address of the host in the private subnet with the public IP address. For incoming traffic, the public IP address is converted back into the private address, and the message is routed to the appropriate host in the private subnet.

Using NAT with the Session Initiation Protocol (SIP) service is more complicated because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. When using NAT with the SIP service, the SIP headers contain information about the caller and the receiver, and the device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media. The device translates SDP information for allocating resources to send and receive the media.

How IP addresses and port numbers in SIP messages are replaced depends on the direction of the message. For an outgoing message, the private IP address and port number of the client are replaced with the public IP address and port number of the Juniper Networks firewall. For an incoming message, the public address of the firewall is replaced with the private address of the client.

When an INVITE message is sent out across the firewall, the SIP Application Layer Gateway (ALG) collects information from the message header into a call table, which it uses to forward subsequent messages to the correct endpoint. When a new message arrives, for example an ACK or 200 OK, the ALG compares the "From:", "To:", and "Call-ID:" fields against the call table to identify the call context of the message. If a new INVITE message arrives that matches the existing call, the ALG processes it as a REINVITE.

When a message containing SDP information arrives, the ALG allocates ports and creates a NAT mapping between them and the ports in the SDP. Because the SDP requires sequential ports for the Real-Time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP) channels, the ALG provides consecutive even-odd ports. If it is unable to find a pair of ports, it discards the SIP message.

This topic contains the following sections:

- Outgoing Calls on page 222
- Incoming Calls on page 222
- Forwarded Calls on page 223
- Call Termination on page 223
- Call Re-INVITE Messages on page 223
- Call Session Timers on page 223
- Call Cancellation on page 223
- Forking on page 224

- SIP Messages on page 224
- SIP Headers on page 224
- SIP Body on page 226
- SIP NAT Scenario on page 226
- Classes of SIP Responses on page 228

Outgoing Calls

When a SIP call is initiated with a SIP request message from the internal to the external network, NAT replaces the IP addresses and port numbers in the SDP and binds the IP addresses and port numbers to the Juniper Networks firewall. Via, Contact, Route, and Record-Route SIP header fields, if present, are also bound to the firewall IP address. The ALG stores these mappings for use in retransmissions and for SIP response messages.

The SIP ALG then opens pinholes in the firewall to allow media through the device on the dynamically assigned ports negotiated based on information in the SDP and the Via, Contact, and Record-Route header fields. The pinholes also allow incoming packets to reach the Contact, Via, and Record-Route IP addresses and ports. When processing return traffic, the ALG inserts the original Contact, Via, Route, and Record-Route SIP fields back into packets.

Incoming Calls

Incoming calls are initiated from the public network to public static NAT addresses or to interface IP addresses on the device. Static NATs are statically configured IP addresses that point to internal hosts; interface IP addresses are dynamically recorded by the ALG as it monitors REGISTER messages sent by internal hosts to the SIP registrar. When the device receives an incoming SIP packet, it sets up a session and forwards the payload of the packet to the SIP ALG.

The ALG examines the SIP request message (initially an INVITE) and, based on information in the SDP, opens gates for outgoing media. When a 200 OK response message arrives, the SIP ALG performs NAT on the IP addresses and ports and opens pinholes in the outbound direction. (The opened gates have a short time-to-live, and they time out if a 200 OK response message is not received quickly.)

When a 200 OK response arrives, the SIP proxy examines the SDP information and reads the IP addresses and port numbers for each media session. The SIP ALG on the device performs NAT on the addresses and port numbers, opens pinholes for outbound traffic, and refreshes the timeout for gates in the inbound direction.

When the ACK arrives for the 200 OK, it also passes through the SIP ALG. If the message contains SDP information, the SIP ALG ensures that the IP addresses and port numbers are not changed from the previous INVITE—if they are, the ALG deletes old pinholes and creates new pinholes to allow media to pass through. The ALG also monitors the Via, Contact, and Record-Route SIP fields and opens new pinholes if it determines that these fields have changed.

Forwarded Calls

A forwarded call is when, for example, user A outside the network calls user B inside the network, and user B forwards the call to user C outside the network. The SIP ALG processes the INVITE from user A as a normal incoming call. But when the ALG examines the forwarded call from B to C outside the network and notices that B and C are reached using the same interface, it does not open pinholes in the firewall, because media will flow directly between user A and user C.

Call Termination

The BYE message terminates a call. When the device receives a BYE message, it translates the header fields just as it does for any other message. But because a BYE message must be acknowledged by the receiver with a 200 OK, the ALG delays call teardown for five seconds to allow time for transmission of the 200 OK.

Call Re-INVITE Messages

Re-INVITE messages add new media sessions to a call and remove existing media sessions. When new media sessions are added to a call, new pinholes are opened in the firewall and new address bindings are created. The process is identical to the original call setup. When one or more media sessions are removed from a call, pinholes are closed and bindings released just as with a BYE message.

Call Session Timers

The SIP ALG uses the Session-Expires value to time out a session if a Re-INVITE or UPDATE message is not received. The ALG gets the Session-Expires value, if present, from the 200 OK response to the INVITE and uses this value for signaling timeout. If the ALG receives another INVITE before the session times out, it resets all timeout values to this new INVITE or to default values, and the process is repeated.

As a precautionary measure, the SIP ALG uses hard timeout values to set the maximum amount of time a call can exist. This ensures that the device is protected should one of the following events occur:

- End systems crash during a call and a BYE message is not received.
- Malicious users never send a BYE in an attempt to attack a SIP ALG.
- Poor implementations of SIP proxy fail to process Record-Route and never send a BYE message.
- Network failures prevent a BYE message from being received.

Call Cancellation

Either party can cancel a call by sending a CANCEL message. Upon receiving a CANCEL message, the SIP ALG closes pinholes through the firewall—if any have been opened—and releases address bindings. Before releasing the resources, the ALG delays the control channel age-out for approximately five seconds to allow time for the final 200 OK to pass through. The call is terminated when the five second timeout expires, regardless of whether a 487 or non-200 response arrives.

Forking

Forking enables a SIP proxy to send a single INVITE message to multiple destinations simultaneously. When the multiple 200 OK response messages arrive for the single call, the SIP ALG parses but updates call information with the first 200 OK messages it receives.

SIP Messages

The SIP message format consists of a SIP header section and the SIP body. In request messages, the first line of the header section is the request line, which includes the method type, request-URI, and protocol version. In response messages, the first line is the status line, which contains a status code. SIP headers contain IP addresses and port numbers used for signaling. The SIP body, separated from the header section by a blank line, is reserved for session description information, which is optional. Junos OS currently supports the SDP only. The SIP body contains IP addresses and port numbers used to transport the media.

SIP Headers

In the following sample SIP request message, NAT replaces the IP addresses in the header fields to hide them from the outside network.

```
INVITE bob@10.150.20.5 SIP/2.0
Via: SIP/2.0/UDP 10.150.20.3:5434
From: alice@10.150.20.3
To: bob@10.150.20.5
Call-ID: a12abcde@10.150.20.3
Contact: alice@10.150.20.3:5434
Route: <sip:netscreen@10.150.20.3:5060>
Record-Route: <sip:netscreen@10.150.20.3:5060>
```

How IP address translation is performed depends on the type and direction of the message. A message can be any of the following:

- Inbound request
- Outbound response
- Outbound request
- Inbound response

Table 29 on page 225 shows how NAT is performed in each of these cases. Note that for several of the header fields the ALG determine more than just whether the messages comes from inside or outside the network. It must also determine what client initiated the call, and whether the message is a request or response.

Table 29: Requesting Messages with NAT Table

Inbound Request (from public to private)	To:	Replace domain with local address
	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	Replace ALG address with local address
	Contact:	None
	Record-Route:	None
	Route:	None
Outbound Response (from private to public)	To:	Replace ALG address with local address
	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	N/A
	Contact:	Replace local address with ALG address
	Record-Route:	Replace local address with ALG address
	Route:	None
Outbound Request (from private to public)	To:	None
	From:	Replace local address with ALG address
	Call-ID:	Replace local address with ALG address
	Via:	Replace local address with ALG address
	Request-URI:	None
	Contact:	Replace local address with ALG address
	Record-Route:	Replace local address with ALG address
	Route:	Replace ALG address with local address

Table 29: Requesting Messages with NAT Table (*continued*)

Outbound Response (from public to private)	To:	None
	From:	Replace ALG address with local address
	Call-ID:	Replace ALG address with local address
	Via:	Replace ALG address with local address
	Request-URI:	N/A
	Contact:	None
	Record-Route:	Replace ALG address with local address
	Route:	Replace ALG address with local address

SIP Body

The SDP information in the SIP body includes IP addresses the ALG uses to create channels for the media stream. Translation of the SDP section also allocates resources, that is, port numbers to send and receive the media.

The following excerpt from a sample SDP section shows the fields that are translated for resource allocation.

```
o=user 2344234 55234434 IN IP4 10.150.20.3
c=IN IP4 10.150.20.3
m=audio 43249 RTP/AVP 0
```

SIP messages can contain more than one media stream. The concept is similar to attaching multiple files to an e-mail message. For example, an INVITE message sent from a SIP client to a SIP server might have the following fields:

```
c=IN IP4 10.123.33.4
m=audio 33445 RTP/AVP 0
c=IN IP4 10.123.33.4
m=audio 33447 RTP/AVP 0
c=IN IP4 10.123.33.4
m=audio 33449 RTP/AVP 0
```

Junos OS supports up to 6 SDP channels negotiated for each direction, for a total of 12 channels per call. For more information, see “SDP Session Descriptions” on page 209.

SIP NAT Scenario

Figure 19 on page 227 and Figure 20 on page 227 show a SIP call INVITE and 200 OK. In Figure 19 on page 227, ph1 sends a SIP INVITE message to ph2. Note how the IP addresses in the header fields—shown in bold font—are translated by the device.

The SDP section of the INVITE message indicates where the caller is willing to receive media. Note that the Media Pinhole contains two port numbers, 52002 and 52003, for RTCP and RTP. The Via/Contact Pinhole provides port number 5060 for SIP signaling.

Observe how, in the 200 OK response message in Figure 20 on page 227, the translations performed in the INVITE message are reversed. The IP addresses in this message, being public, are not translated, but gates are opened to allow the media stream access to the private network.

Figure 19: SIP NAT Scenario 1

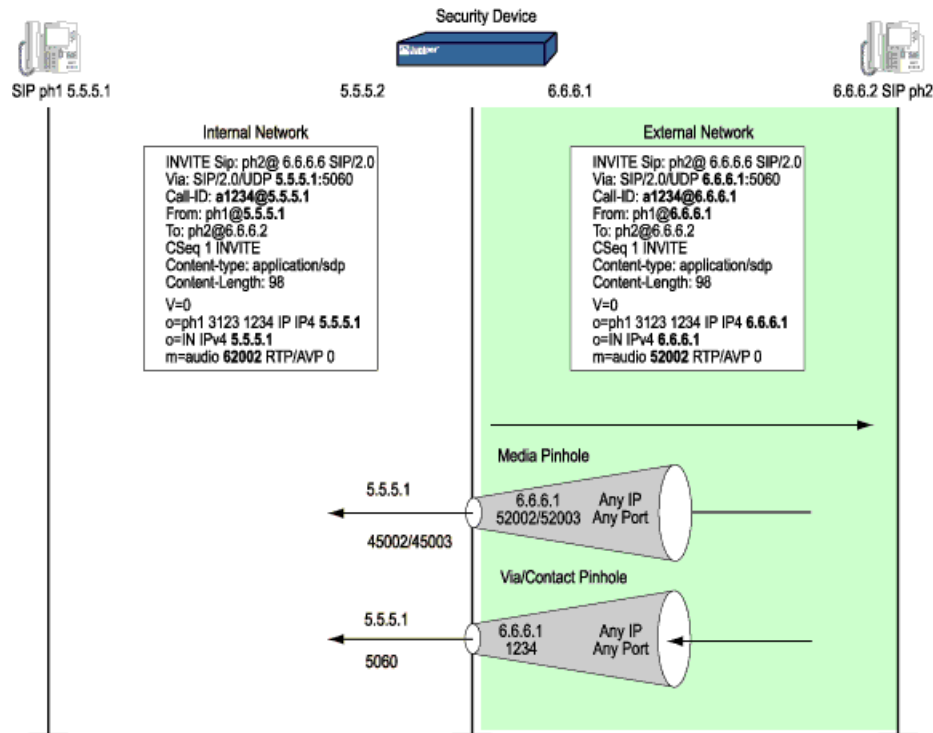
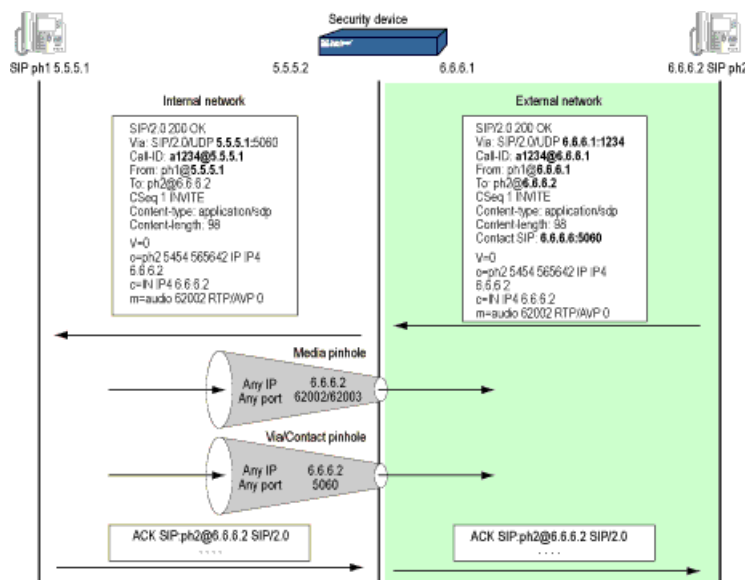


Figure 20: SIP NAT Scenario 2



Classes of SIP Responses

SIP responses provide status information about SIP transactions and include a response code and a reason phrase. SIP responses are grouped into the following classes:

- Informational (100 to 199)—Request received, continuing to process the request.
- Success (200 to 299)—Action successfully received, understood, and accepted.
- Redirection (300 to 399)—Further action required to complete the request.
- Client Error (400 to 499)—Request contains bad syntax or cannot be fulfilled at this server.
- Server Error (500 to 599)—Server failed to fulfill an apparently valid request.
- Global Failure (600 to 699)—Request cannot be fulfilled at any server.

Table 30 on page 228 provides a complete list of current SIP responses.

Table 30: SIP Responses

Informational	100 Trying	180 Ringing	181 Call is being forwarded
	182 Queued	183 Session progress	
Success	200 OK	202 Accepted	
Redirection	300 Multiple choices	301 Moved permanently	302 Moved temporarily
	305 Use proxy	380 Alternative service	
Client Error	400 Bad request	401 Unauthorized	402 Payment required
	403 Forbidden	404 Not found	405 Method not allowed
	406 Not acceptable	407 Proxy authentication required	408 Request time-out
	409 Conflict	410 Gone	411 Length required
	413 Request entity too large	414 Request URL too large	415 Unsupported media type
	420 Bad extension	480 Temporarily not available	481 Call leg/transaction does not exist
	482 Loop detected	483 Too many hops	484 Address incomplete
	485 Ambiguous	486 Busy here	487 Request canceled
	488 Not acceptable here		

Table 30: SIP Responses (*continued*)

Server Error	500 Server internal error	501 Not implemented	502 Bad gateway
	502 Service unavailable	504 Gateway time-out	505 SIP version not supported
Global Failure	600 Busy everywhere	603 Decline	604 Does not exist anywhere
	606 Not acceptable		

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALGs on page 207
 - Understanding Incoming SIP ALG Call Support Using the SIP Registrar and NAT on page 229
 - Example: Configuring Interface Source NAT for Incoming SIP Calls (CLI) on page 230
 - Example: Configuring a Source NAT Pool for Incoming SIP Calls (CLI) on page 232
 - Example: Configuring Static NAT for Incoming SIP Calls (CLI) on page 234
 - Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone (CLI) on page 235
 - Example: Configuring the SIP Proxy and NAT in the Public Zone (CLI) on page 237
 - Example: Configuring a Three-Zone SIP ALG and NAT Scenario (CLI) on page 238

Understanding Incoming SIP ALG Call Support Using the SIP Registrar and NAT

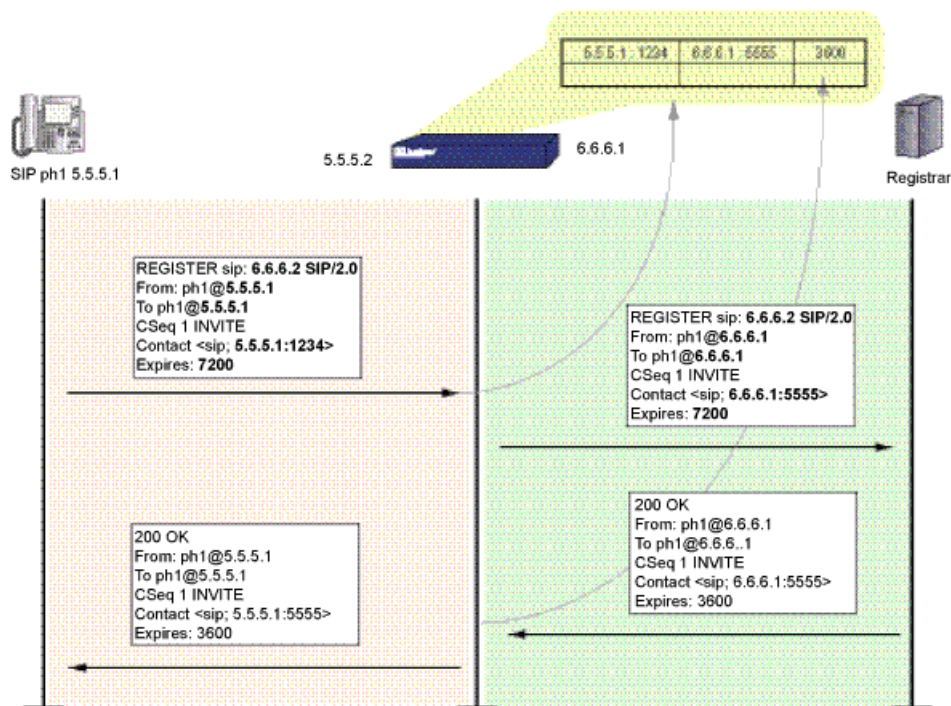
Session Initiation Protocol (SIP) registration provides a discovery capability by which SIP proxies and location servers can identify the location or locations where users want to be contacted. A user registers one or more contact locations by sending a REGISTER message to the registrar. The To and Contact fields in the REGISTER message contain the address-of-record Uniform Resource Identifier (URI) and one or more contact URIs, as shown in Figure 21 on page 230. Registration creates bindings in a location service that associates the address-of-record with the contact address or addresses.

The device monitors outgoing REGISTER messages, performs Network Address Translation (NAT) on these addresses, and stores the information in an Incoming NAT table. Then, when an INVITE message is received from outside the network, the device uses the Incoming NAT table to identify which internal host to route the INVITE message to. You can take advantage of SIP proxy registration service to allow incoming calls by configuring interface source NAT or NAT pools on the egress interface of the device. Interface source NAT is adequate for handling incoming calls in a small office, whereas we recommend setting up source NAT pools for larger networks or an enterprise environment.



NOTE: Incoming call support using interface source NAT or a source NAT pool is supported for SIP and H.323 services only. For incoming calls, Junos OS currently supports UDP and TCP only. Domain name resolution is also currently not supported; therefore, URIs must contain IP addresses, as shown in Figure 21 on page 230.

Figure 21: Using the SIP Registrar



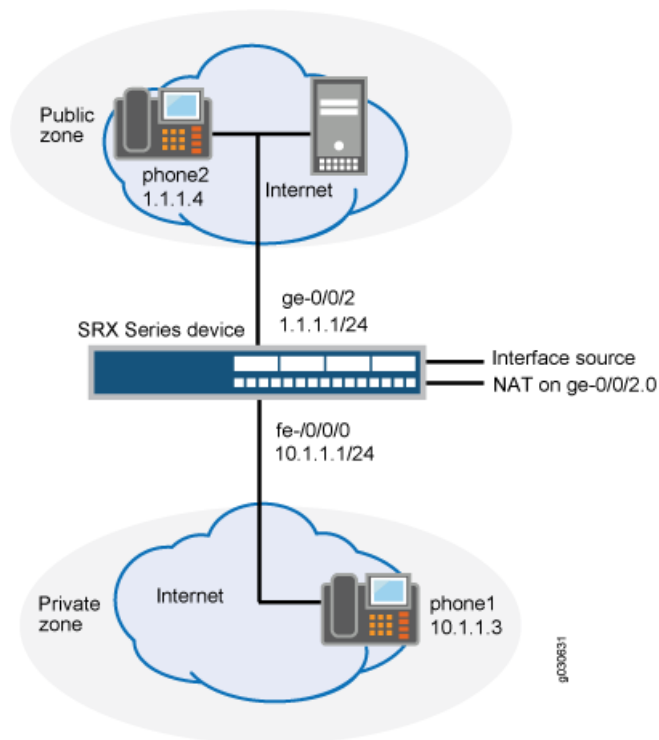
- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - ALG Overview on page 169
 - Understanding SIP ALGs and NAT on page 221
 - SIP ALG Configuration Overview on page 213

Example: Configuring Interface Source NAT for Incoming SIP Calls (CLI)

In a two-zone scenario with the SIP proxy server in an external, or public zone, you can use NAT for incoming calls by configuring source NAT on the interface to the public zone.

In this example, phone1 is on the ge-0/0/0 interface in the private zone, and phone2 and the proxy server are on the ge-0/0/2 interface in the public zone. You configure interface source NAT on ge-0/0/2.0 for incoming calls, then create a policy permitting SIP traffic from the public zone to the private zone and reference the source NAT in the policy. You also create a policy that permits SIP traffic from the private to the public zone, again referencing the source NAT address pool. This enables phone1 in the private zone to register with the proxy in the public zone. See Figure 22 on page 231.

Figure 22: Source NAT for Incoming Calls



To configure interface source NAT for incoming calls:

1. Configure interfaces.


```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
user@host# set security zones security-zone private interface ge-0/0/0.0
```
2. Configure addresses.


```
user@host# set security zones security-zone private address-book address phone1
10.1.1.3/32
user@host# set security zones security-zone public address-book address proxy
10.1.1.3/32
user@host# set security zones security-zone public address-book address phone2
1.1.1.4/32
```
3. Configure zones.


```
user@host# set security zones security-zone private
user@host# set security zones security-zone public
user@host# set security zones security-zone private interfaces ge-0/0/0.0
user@host# set security zones security-zone public interfaces ge-0/0/2.0
```
4. Configure source NAT.


```
user@host# set security nat interface ge-0/0/2.0 source-nat allow-incoming
user@host# set security nat source-nat address-persistent
```
5. Configure policies.

```
user@host# set security policies from-zone private to-zone public policy outgoing
match source-address phone1 destination-address any application junos-sip
user@host# set security policies from-zone private to-zone public policy outgoing
then permit source-nat interface
user@host# set security policies from-zone public to-zone private policy incoming
match source-address any destination-address incoming-nat-fe0/0/2.0 application
junos-sip
user@host# set security policies from-zone public to-zone private policy incoming
then permit
```

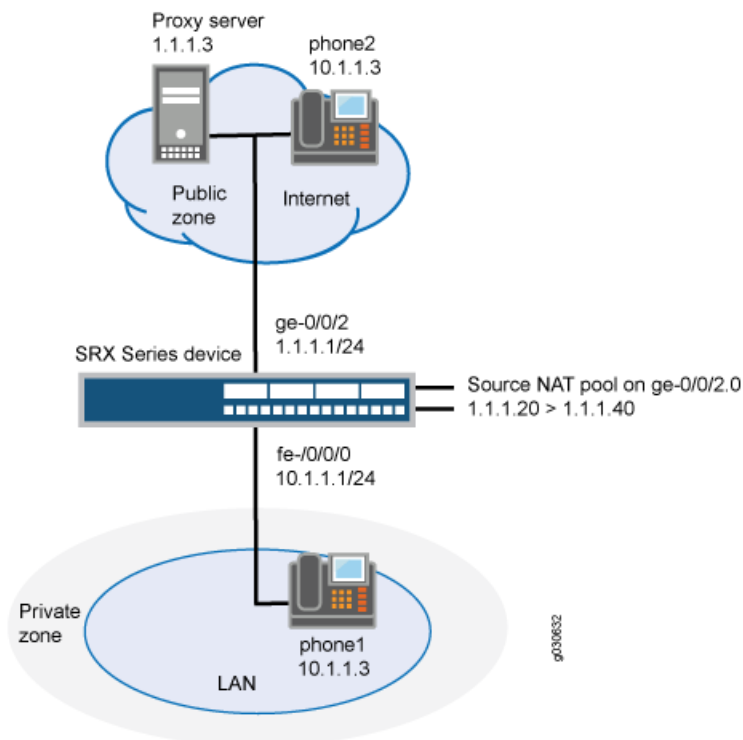
- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALGs and NAT on page 221
 - SIP ALG Configuration Overview on page 213
 - Verifying SIP ALG Configurations on page 241

Example: Configuring a Source NAT Pool for Incoming SIP Calls (CLI)

In a two-zone scenario with the Session Initiation Protocol (SIP) proxy server in an external, or public zone, you can use Network Address Translation (NAT) for incoming calls by configuring a NAT pool on the interface to the public zone.

In this example, phone1 is in the private zone, and phone2 and the proxy server are in the public zone. You configure a source NAT pool on the ge-0/0/2.0 interface to do NAT on incoming calls, then set a policy permitting SIP traffic from the public zone to the private zone and reference the NAT pool in the policy. You also create a policy that permits SIP traffic from the private to the public zone. This enables phone1 in the private zone to register with the proxy in the public zone. See Figure 23 on page 233.

Figure 23: Source NAT Pool for Incoming Calls



To configure a source NAT pool for incoming calls:

1. Configure interfaces.


```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
user@host# set security zones security-zone private interface ge-0/0/0.0
```
2. Configure addresses.


```
user@host# set security zones security-zone private address-book address phone1
10.1.1.3/32
user@host# set security zones security-zone public address-book address proxy
10.1.1.3/32
user@host# set security zones security-zone public address-book address phone2
1.1.1.4/32
```
3. Configure zones.


```
user@host# set security zones security-zone private
user@host# set security zones security-zone public
user@host# set security zones security-zone private interfaces ge-0/0/0.0
user@host# set security zones security-zone public interfaces ge-0/0/2.0
```
4. Configure the source NAT pool.


```
user@host# set security nat source-nat address-persistent
user@host# set security nat interface ge-0/0/2.0 source-nat pool sip-pool
address-range low 1.1.1.20 high 1.1.1.60
user@host# set security nat interface ge-0/0/2.0 source-nat pool sip-pool allow
incoming
```

5. Configure policies.

```

user@host# set security policies from-zone private to-zone public policy outgoing
match source-address phone1 destination-address any application junos-sip
user@host# set security policies from-zone private to-zone public policy outgoing
then permit source-nat pool sip-pool
user@host# set security policies from-zone private to-zone public policy incoming
match source-address any destination-address incoming-nat-sip-pool application
junos-sip
user@host# set security policies from-zone public to-zone private policy incoming
then permit

```

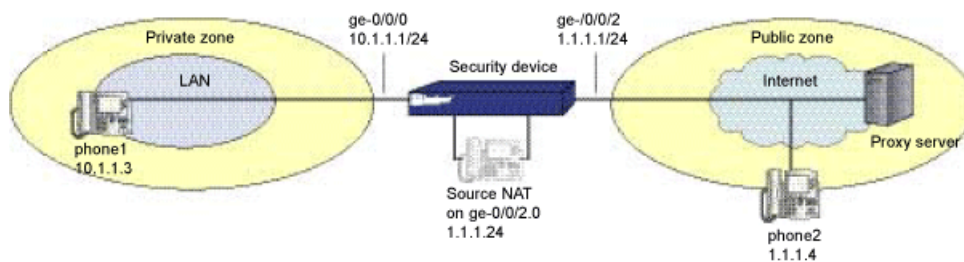
- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALGs and NAT on page 221
 - SIP ALG Configuration Overview on page 213
 - Verifying SIP ALG Configurations on page 241

Example: Configuring Static NAT for Incoming SIP Calls (CLI)

When you locate the SIP proxy server in an external, or public, zone, static NAT configured on the interface to the public will enable callers in the internal, or private, zone to register with the proxy.

In this example, phone1 is on the ge-0/0/0 interface in the private zone, and phone2 and the proxy server are on the ge-0/0/2 interface in the public zone. You configure static NAT on the ge-0/0/2.0 interface to phone1, then create policies that allow SIP traffic from the public zone to the private zone, and reference the static NAT in the policy. This example is similar to the “Example: Configuring Interface Source NAT for Incoming SIP Calls (CLI)” on page 230 and “Example: Configuring a Source NAT Pool for Incoming SIP Calls (CLI)” on page 232, except that with static NAT you need one public address for each private address in the private zone, while with a DIP pool a single interface address can serve multiple private addresses. See Figure 24 on page 234.

Figure 24: Static NAT for Incoming Calls



To configure static NAT for incoming calls:

1. Configure interfaces.

```

user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
user@host# set security zones security-zone private interface ge-0/0/0.0

```

2. Configure addresses.

```

user@host# set security zones security-zone private address-book address phone1
10.1.1.3/32
user@host# set security zones security-zone public address-book address proxy
10.1.1.3/32
user@host# set security zones security-zone public address-book address phone2
1.1.1.4/32

```

3. Configure zones.

```

user@host# set security zones security-zone private
user@host# set security zones security-zone public
user@host# set security zones security-zone private interfaces ge-0/0/0.0
user@host# set security zones security-zone public interfaces ge-0/0/2.0

```

4. Configure static NAT.

```

user@host# set security nat interface ge-0/0/2.0 static-nat 1.1.1.3/32 host 10.1.1.3/32

```

5. Configure policies.

```

user@host# set security policies from-zone public to-zone private policy incoming
match source-address any destination-address static_nat_1.1.1.3-32 application
junos-jsrp
user@host# set security policies from-zone public to-zone private policy incoming
then permit

```

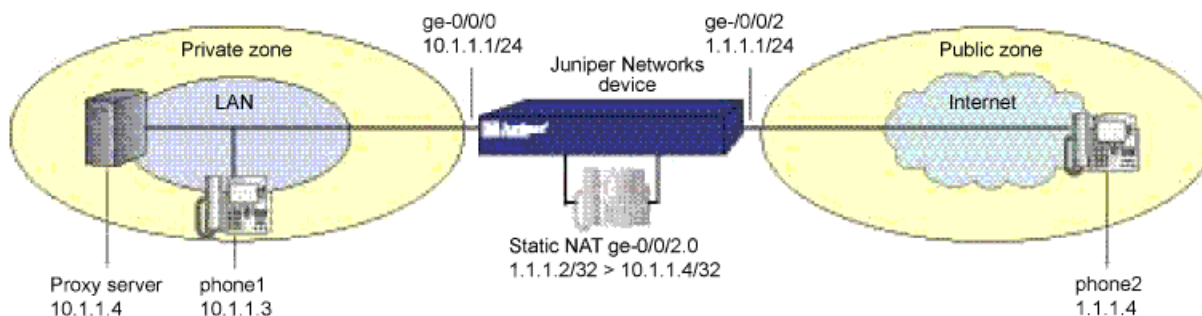
- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALGs and NAT on page 221
 - SIP ALG Configuration Overview on page 213
 - Verifying SIP ALG Configurations on page 241

Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone (CLI)

With the SIP proxy server in the internal, or private, zone, static NAT on the interface to the external, or public, zone is sufficient to allow callers in the public zone to register with the proxy server.

In this example, phone1 and the SIP proxy server are on the ge-0/0/0 interface in the private zone, and phone2 is on the ge-0/0/2 interface in the public zone. You configure static NAT on the ge-0/0/2 interface to the proxy server to allow phone2 to register with the proxy, then create a policy allowing SIP traffic from the public to the private zone to enable callers in the public zone to register with the proxy, and a policy from the private to the public zone to allow phone1 to call out. See Figure 25 on page 236.

Figure 25: Proxy in the Private Zone



To configure the SIP proxy in the private zone and NAT in the public zone:

1. Configure interfaces.

```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
```

2. Configure zones.

```
user@host# set security zones security-zone private
user@host# set security zones security-zone public
user@host# set security zones security-zone private interfaces ge-0/0/0.0
user@host# set security zones security-zone public interfaces ge-0/0/2.0
```

3. Configure addresses.

```
user@host# set security zones security-zone private address-book address phone1
10.1.1.3/32
user@host# set security zones security-zone private address-book address proxy
10.1.1.4/32
user@host# set security zones security-zone public address-book address phone2
1.1.1.4/32
```

4. Configure static NAT.

```
user@host# set security nat interface ge-0/0/2.0 static-nat 1.1.1.2/32 host 10.1.1.4/32
```

5. Configure policies.

```
user@host# set security policies from-zone private to-zone public policy outgoing
match source-address any
user@host# set security policies from-zone private to-zone public policy outgoing
match destination-address phone2
user@host# set security policies from-zone private to-zone public policy outgoing
match application junos-sip
user@host# set security policies from-zone private to-zone public policy outgoing
then permit source-nat interface
user@host# set security policies from-zone public to-zone private policy incoming
match source-address phone2
user@host# set security policies from-zone public to-zone private policy incoming
match destination-address static_nat_1.1.1.2_32
user@host# set security policies from-zone public to-zone private policy incoming
match application junos-sip
user@host# set security policies from-zone public to-zone private policy incoming
then permit
```

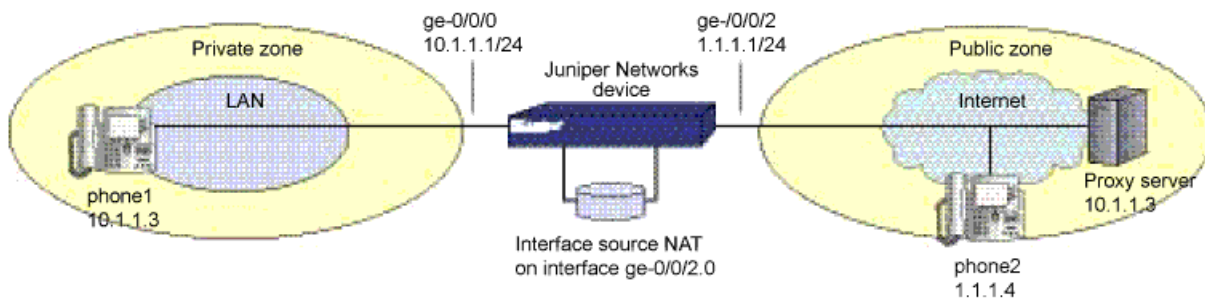
- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALGs and NAT on page 221
 - SIP ALG Configuration Overview on page 213
 - Verifying SIP ALG Configurations on page 241

Example: Configuring the SIP Proxy and NAT in the Public Zone (CLI)

When you locate the SIP proxy server in an external, or public, zone, you will typically want to configure NAT on the interface to that zone.

In this example, phone1 is on the ge-0/0/0.0 interface in the private zone, and the proxy server and phone2 are on the ge-0/0/2.0 interface in the public zone. You configure source NAT on the ge-0/0/2.0 interface in the public zone, then create a policy permitting SIP traffic from the public zone to the private zone and reference the NAT interface. You also create a policy from private to public to allow phone1 to register with the proxy server in the public zone. See Figure 26 on page 237.

Figure 26: Proxy in the Public Zone



To configure the SIP proxy in the public zone:

1. Configure interfaces.


```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 1.1.1/24
```
2. Configure zones.


```
user@host# set security zones security-zone private
user@host# set security zones security-zone public
user@host# set security zones security-zone private interfaces ge-0/0/0.0
user@host# set security zones security-zone public interfaces ge-0/0/2.0
```
3. Configure addresses.


```
user@host# set security zones security-zone private address-book address phone1
10.1.1.3/32
user@host# set security zones security-zone public address-book address phone2
1.1.1.4/32
user@host# set security zones security-zone public address-book address proxy
1.1.1.3/32
```

4. Configure interface source NAT.

```
user@host# set security nat source-nat address-persistent
user@host# set security nat interface ge-0/0/2.0 allow-incoming
```

5. Configure policies.

```
user@host# set security policies from-zone private to-zone public policy outgoing
match source-address phone1
user@host# set security policies from-zone private to-zone public policy outgoing
match destination-address any
user@host# set security policies from-zone private to-zone public policy outgoing
match application junos-sip
user@host# set security policies from-zone private to-zone public policy outgoing
then permit source-nat interface
user@host# set security policies from-zone public to-zone private policy incoming
match source-address any
user@host# set security policies from-zone public to-zone private policy incoming
match destination-address incoming_nat_ge-0/0/2.0
user@host# set security policies from-zone public to-zone private policy incoming
match application junos-sip
user@host# set security policies from-zone public to-zone private policy incoming
then permit
```

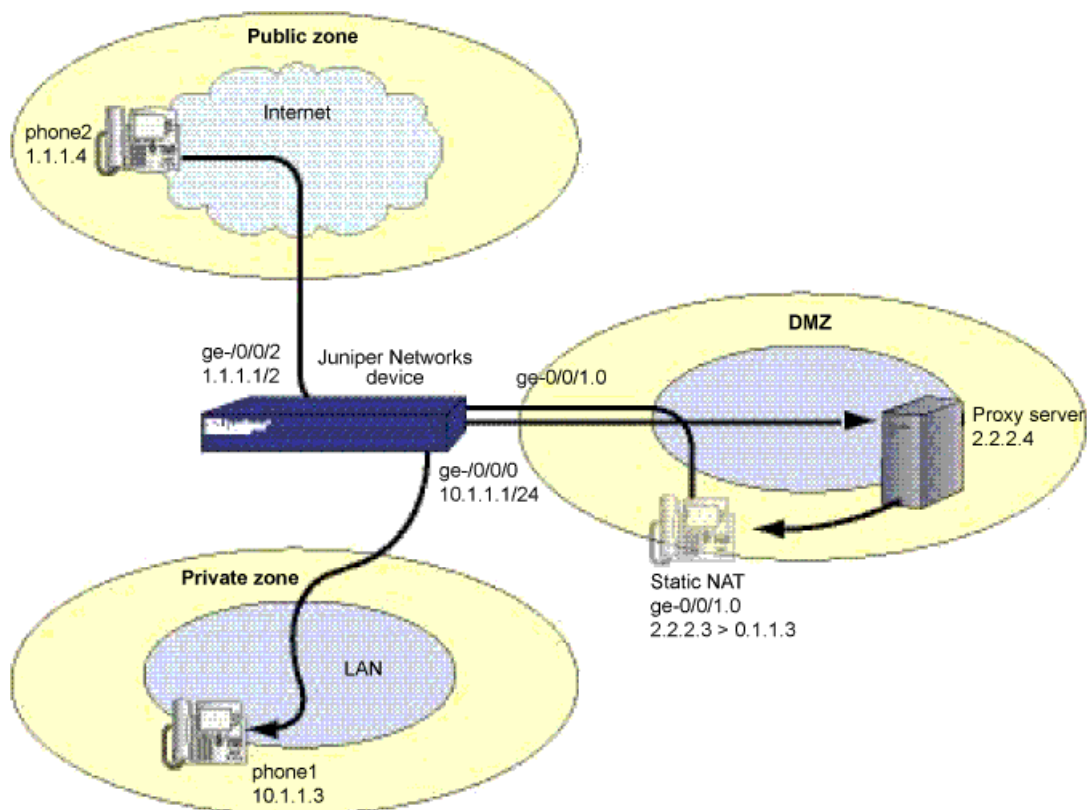
- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALGs and NAT on page 221
 - SIP ALG Configuration Overview on page 213
 - Verifying SIP ALG Configurations on page 241

Example: Configuring a Three-Zone SIP ALG and NAT Scenario (CLI)

In a three-zone SIP configuration, the SIP proxy server is typically in a different zone than the calling and called parties. Such a scenario requires additional address and zone configuration, and policies to ensure that all parties have access to each other and to the proxy server.

In this example, phone1 is on the ge-0/0/0 interface in the private zone, phone2 is on the ge-0/0/2 interface in the public zone, and the proxy server is on the ge-0/0/1.0 interface in the DMZ. You configure static NAT on the ge-0/0/1 interface to phone1 in the private zone. You then create policies from the private zone to the DMZ and from the DMZ to the private zone, from the public zone to the DMZ and from the DMZ to the public zone, and from the private zone to the public zone. The arrows in Figure 27 on page 239 show the flow of SIP signaling traffic when phone2 in the public zone places a call to phone1 in the private zone. After the session is initiated, the media flows directly between phone1 and phone2.

Figure 27: Three-Zone SIP Configuration with Proxy in the DMZ



To configure a three-zone SIP scenario:

1. Configure interfaces.


```

user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 2.2.2.2/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 1.1.1.1/24
      
```
2. Configure zones.


```

user@host# set security zones security-zone private interfaces ge-0/0/0.0
user@host# set security zones security-zone public interfaces ge-0/0/2.0
user@host# set security zones security-zone dmz interfaces ge-0/0/1.0
      
```
3. Configure addresses.


```

user@host# set security zones security-zone private address-book address phone1
10.1.1.3/32
user@host# set security zones security-zone public address-book address phone2
1.1.1.4/32
user@host# set security zones security-zone dmz address-book address proxy
2.2.2.4/32
      
```
4. Configure static NAT.


```

set security nat static rule-set incoming-SIP from zone dmz
set security nat static rule-set incoming-SIP rule phone1 match destination-address
2.2.2.3/32
      
```

```
set security nat static rule-set incoming-SIP rule phone1 then static-nat prefix 10.1.1.3/32
```

5. Configure interface NAT for communication from phone1 to proxy.

```
set security nat source rule-set sip-phones from zone private
set security nat source rule-set sip-phones to zone dmz
set security nat source rule-set sip-phones rule phone1 match source-address 10.1.1.3/32
set security nat source rule-set sip-phones rule phone1 then source-nat interface
```

6. Configure interface NAT for communication from phone1 to phone2:

```
set security nat source rule-set sip-phones from zone private
set security nat source rule-set sip-phones to zone public
set security nat source rule-set sip-phones rule phone1 match source-address 10.1.1.3/32
set security nat source rule-set sip-phones rule phone1 then source-nat interface
```

7. Configure policies.

```
user@host# set security policies from-zone private to-zone dmz policy private-to-proxy
match source-address phone1
user@host# set security policies from-zone private to-zone dmz policy private-to-proxy
match destination-address proxy
user@host# set security policies from-zone private to-zone dmz policy private-to-proxy
match application junos-sip
user@host# set security policies from-zone private to-zone dmz policy private-to-proxy
then permit source-nat interface
user@host# set security policies from-zone public to-zone dmz policy public-to-proxy
match source-address phone2
user@host# set security policies from-zone public to-zone dmz policy public-to-proxy
match destination-address proxy
user@host# set security policies from-zone public to-zone dmz policy public-to-proxy
match application junos-sip
user@host# set security policies from-zone public to-zone dmz policy public-to-proxy
then permit
user@host# set security policies from-zone private to-zone public policy
private-to-public match source-address phone1
user@host# set security policies from-zone private to-zone public policy
private-to-public match destination-address phone2
user@host# set security policies from-zone private to-zone public policy
private-to-public match application junos-sip
user@host# set security policies from-zone private to-zone public policy
private-to-public then permit source-nat interface
user@host# set security policies from-zone dmz to-zone private policy proxy-to-private
match source-address proxy
user@host# set security policies from-zone dmz to-zone private policy proxy-to-private
match destination-address static_nat_2.2.2.3_32
user@host# set security policies from-zone dmz to-zone private policy proxy-to-private
match application junos-sip
user@host# set security policies from-zone dmz to-zone private policy proxy-to-private
then permit
user@host# set security policies from-zone dmz to-zone public policy proxy-to-public
match source-address proxy
user@host# set security policies from-zone dmz to-zone public policy proxy-to-public
match destination-address phone2
user@host# set security policies from-zone dmz to-zone public policy proxy-to-public
match application junos-sip
user@host# set security policies from-zone dmz to-zone public policy proxy-to-public
then permit
```


- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SIP ALGs and NAT on page 221
 - SIP ALG Configuration Overview on page 213
 - Verifying SIP ALG Configurations on page 241

Verifying SIP ALG Configurations

- Verifying SIP ALGs on page 241
- Verifying SIP ALG Calls on page 241
- Verifying SIP ALG Call Details on page 242
- Verifying SIP ALG Counters on page 242
- Verifying the Rate of SIP ALG Messages on page 243

Verifying SIP ALGs

Purpose Verify SIP ALG verification options.

Action From the CLI, enter the **show security alg sip ?** command.

```
user@host> show security alg sip ?
Possible completions:
  calls           Show SIP calls
  counters        Show SIP counters
  rate            Show SIP rate
```

Meaning The output shows a list of all SIP verification parameters. Verify the following information:

- Calls—Lists all SIP calls.
- Counters—Provides counters of response codes for each SIP request method and error type.
- Rate—Provides speed and periodicity of SIP signaling messages.

Verifying SIP ALG Calls

Purpose Display information about active calls.

Action From the J-Web interface, select **Monitor>ALGs>SIP>Calls**. Alternatively, from the CLI, enter the **show security alg sip calls** command.

```
user@host> show security alg sip calls
Total number of calls: 1
  Call ID: 47090a32@30.2.20.5
  Method: INVITE
```

Meaning The output shows a list of all active SIP calls. Verify the User Agent Server (UAS) call ID and local and remote tags, and the state of the call.

Verifying SIP ALG Call Details

Purpose Display address and SDP about active calls.

Action From the J-Web interface, select **Monitor>ALGs>SIP>Details**. Alternatively, from the CLI, enter the **show security alg sip calls detail** command.

```
user@host> show security alg sip calls detail
Total number of calls: 1
  Call ID    : 47090a32@30.2.20.5
Method      : INVITE
State       : SETUP
Group ID    : 24575
```

Meaning The output provides details about all active SIP calls. Verify the following information:

- The total number of calls, their ID and tag information, and state
- Remote group ID
- The IP addresses and port numbers and SDP connection and media details

Verifying SIP ALG Counters

Purpose Display information about SIP counters.

Action From the J-Web interface, select **Monitor>ALGs>SIP>Counters**. Alternatively, from the CLI, enter the **show security alg sip counters** command.

```
user@host> show security alg sip counters
Method      T      1xx      2xx      3xx      4xx      5xx      6xx
            RT      RT      RT      RT      RT      RT      RT
INVITE      4       4       3       0       0       0       0
            0       0       0       0       0       0       0
CANCEL      0       0       0       0       0       0       0
            0       0       0       0       0       0       0
ACK         3       0       0       0       0       0       0
            0       0       0       0       0       0       0
BYE         3       0       3       0       0       0       0
            0       0       0       0       0       0       0
REGISTER    7       0       7       0       0       0       0
            0       0       0       0       0       0       0
OPTIONS     0       0       0       0       0       0       0
            0       0       0       0       0       0       0
INFO        0       0       0       0       0       0       0
            0       0       0       0       0       0       0
MESSAGE     0       0       0       0       0       0       0
            0       0       0       0       0       0       0
NOTIFY      0       0       0       0       0       0       0
            0       0       0       0       0       0       0
PRACK       0       0       0       0       0       0       0
            0       0       0       0       0       0       0
PUBLISH     0       0       0       0       0       0       0
            0       0       0       0       0       0       0
REFER       0       0       0       0       0       0       0
            0       0       0       0       0       0       0
SUBSCRIBE   0       0       0       0       0       0       0
            0       0       0       0       0       0       0
```

UPDATE	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
BENOTIFY	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
SERVICE	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
OTHER	0	0	0	0	0	0	0
	0	0	0	0	0	0	0

SIP Error Counters

```

-----
Total Pkt-in                               :34
Total Pkt dropped on error                 :0
      Call error                           :0
IP resolve error                           :0
NAT error                                  :0
Resource manager error                     :0
RR header exceeded max                     :0
Contact header exceeded max                :0
Call Dropped due to limit                  :0
SIP stack error                            : 0
SIP decode error                           : 0
SIP unknown method error                   : 0
RTO message sent                           : 0
RTO message received                       : 0
RTO buffer allocation failure               : 0
RTO buffer transmit failure                 : 0
RTO send processing error                   : 0
RTO receive processing error                : 0
RTO receive invalid length                  : 0
RTO receive call process error              : 0
RTO receive call allocation error           : 0
RTO receive call register error            : 0
RTO receive invalid status error            : 0

```

Meaning The output provides a count of all SIP response codes transmitted and received, and of SIP errors. Verify the following information:

- A count of transmissions of response codes for each SIP request method
- A count of all possible error types

Verifying the Rate of SIP ALG Messages

Purpose Display information about SIP message rate.

Action From the J-Web interface, select **Monitor>ALGs>SIP>Rate**. Alternatively, from the CLI, enter the **show security alg sip rate** command.

```

user@host> show security alg sip rate
CPU ticks per microseconds is 3735928559
Time taken for the last message is 0 microseconds
Total time taken for 0 messages is 0 microseconds(in less than 10 minutes)
Rate: 3735928559 messages/second

```

Meaning The output provides information about CPU usage for messages, and speed and periodicity of SIP signaling messages. Verify the following information:

- CPU ticks per US

- Passage time for last message, for all messages, and the rate at which messages transit the network

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- SIP ALG Configuration Overview on page 213
- Example: Configuring Interface Source NAT for Incoming SIP Calls (CLI) on page 230
- Example: Configuring a Source NAT Pool for Incoming SIP Calls (CLI) on page 232
- Example: Configuring Static NAT for Incoming SIP Calls (CLI) on page 234
- Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone (CLI) on page 235
- Example: Configuring the SIP Proxy and NAT in the Public Zone (CLI) on page 237
- Example: Configuring a Three-Zone SIP ALG and NAT Scenario (CLI) on page 238

CHAPTER 13

SCCP ALGs

- Understanding SCCP ALGs on page 245
- SCCP ALG Configuration Overview on page 250
- SCCP ALG Inactive Media Timeout on page 251
- SCCP ALG Unknown Message Types on page 252
- SCCP ALG DoS Attack Protection on page 254
- Example: Configuring the SCCP ALG CallManager/TFTP Server in the Private Zone (CLI) on page 256
- Verifying SCCP ALG Configurations on page 257

Understanding SCCP ALGs

The Skinny Client Control Protocol (SCCP) is a Cisco proprietary protocol for call signaling. Skinny is based on a call-agent-based call-control architecture. The control protocol uses binary-coded frames encoded on TCP frames sent to well-known TCP port number destinations to set up and tear down RTP media sessions.

The SCCP protocol just as other call control protocols, negotiates media endpoint parameters—specifically the Real-Time Transport Protocol (RTP) port number and the IP address of media termination—by embedding information in the control packets. The SCCP Application Layer Gateway (ALG) parses these control packets and facilitates media and control packets to flow through the system.

The SCCP ALG also implements rate limiting of calls and helps protect critical resources from overloading and denial-of-service (DoS) attacks.

The following functions are implemented by the SCCP ALG in Junos OS:

- Validation of SCCP protocol data units
- Translation of embedded IP address and port numbers
- Allocation of firewall resources (pinholes and gates) to pass media
- Aging out idle calls
- Configuration API for SCCP ALG parameters
- Operational mode API for displaying counters, status and statistics

In the SCCP architecture, a proxy, known as the CallManager, does most of the processing. IP phones, also called End Stations, run the SCCP client and connect to a primary (and, if available, a secondary) CallManager over TCP on port 2000 and register with the primary CallManager. This connection is then used to establish calls coming to or from the client.

The SCCP ALG supports the following:

- Call flow from a SCCP client, through the CallManager, to another SCCP client.
- Seamless failover—Switches over all calls in process to the standby firewall during failure of the primary.
- Voice-over-IP (VoIP) signaling payload inspection—Fully inspects the payload of incoming VoIP signaling packets. Any malformed packet attack is blocked by the ALG.
- SCCP signaling payload inspection—Fully inspects the payload of incoming SCCP signaling packets. Any malformed-packet attack is blocked by the ALG.
- Stateful processing—Invokes the corresponding VoIP-based state machines to process the parsed information. Any out-of-state or out-of-transaction packet is identified and properly handled.
- Network Address Translation (NAT)—Translates any embedded IP address and port information in the payload, based on the existing routing information and network topology, with the translated IP address and port number, if necessary.
- Pinhole creation and management for VoIP traffic—Identifies IP address and port information used for media or signaling and dynamically opens (and closes) pinholes to securely stream the media.

This topic includes the following sections:

- SCCP Security on page 246
- SCCP Components on page 247
- SCCP Transactions on page 247
- SCCP Control Messages and RTP Flow on page 248
- SCCP Messages on page 249

SCCP Security

The SCCP ALG includes the following security features:

- Stateful inspection of SCCP control messages over TCP and validation of the message format, and message validity for the current call state. Invalid messages are dropped.
- Security policy enforcement between Cisco IP phones and Cisco CallManager.
- Protect against call flooding by rate limiting the number of calls processed by the ALG.
- Seamless failover of calls, including the ones in progress in case of device failure in a clustered deployment.

SCCP Components

The principal components of the SCCP VoIP architecture include the following:

- SCCP Client on page 247
- CallManager on page 247
- Cluster on page 247

SCCP Client

The SCCP client runs on an IP phone, also called an *End Station*, which uses SCCP for signaling and for making calls. For an SCCP client to make a call, it must first register with a Primary CallManager (and a secondary, if available). The connection between the client and the CallManager is over TCP on port 2000. This connection is then used to establish calls to or from the client. Transmission of media is over RTP, UDP, and IP.

CallManager

The CallManager implements SCCP call control server software and has overall control of all devices and communication in the SCCP VoIP network. Its functions include defining, monitoring and controlling SCCP groups, regions of numbers, and route plans; providing initialization, admission, and registration of devices on the network; providing a redundant database that contains addresses, phone numbers, and number formats; and initiating contact with called devices or their agents to establish logical sessions in which voice communication can flow.

Cluster

A *cluster* is a collection of SCCP clients and a CallManager. The CallManager in the cluster detects all SCCP clients in the cluster. There can be more than one CallManager for backup in a cluster. CallManager behavior varies in each of the following cluster scenarios:

- Intra-Cluster, in which the CallManager detects each SCCP client, and the call is between SCCP clients of the same cluster.
- Inter-Cluster, in which the CallManager needs to communicate with another CallManager using H.323 for call setup.
- Inter-Cluster calls using the gatekeeper for admission control and address resolution.

CallManager behavior also varies with calls between an SCCP client and a phone in a public switched telephone network (PSTN), and with calls between an SCCP client and a phone in another administrative domain that is using H.323.

SCCP Transactions

SCCP transactions are the processes that need to take place in order for an SCCP call to proceed. SCCP transactions include the following processes:

- Client Initialization on page 248
- Client Registration on page 248
- Call Setup on page 248
- Media Setup on page 248

Client Initialization

To initialize, the SCCP client needs to determine the IP address of the CallManager, its own IP address, and other information about the IP gateway and DNS servers. Initialization takes place on the local LAN. The client sends a Dynamic Host Control Protocol (DHCP) request to get an IP address, the DNS server address, and the TFTP server name and address. The client needs the TFTP server name to download the configuration file called *sepmacaddr.cnf*. If the TFTP name is not given, the client uses the default filename in the IP phone. The client then downloads the .cnf (xml) configuration file from TFTP server. CNF files contain the IP address or addresses of the primary and secondary Cisco CallManager. With this information, the client contacts the CallManager to register.

Client Registration

The SCCP client, after initialization, registers with the CallManager over a TCP connection on well-known default port 2000. The client registers by providing the CallManager with its IP address, the MAC address of the phone, and other information, such as protocol and version. The client cannot initiate or receive calls until it is registered. Keepalive messages keep this TCP connection open between the client and CallManager so that the client can initiate or receive calls at any time, provided that a policy on the device allows this.

Call Setup

IP phone-to-IP phone call setup using SCCP is always handled by the CallManager. Messages for call setup are sent to the CallManager, which returns messages appropriate to the status of the call. If call setup is successful, and a policy on the device allows the call, the CallManager sends the media setup messages to the client.

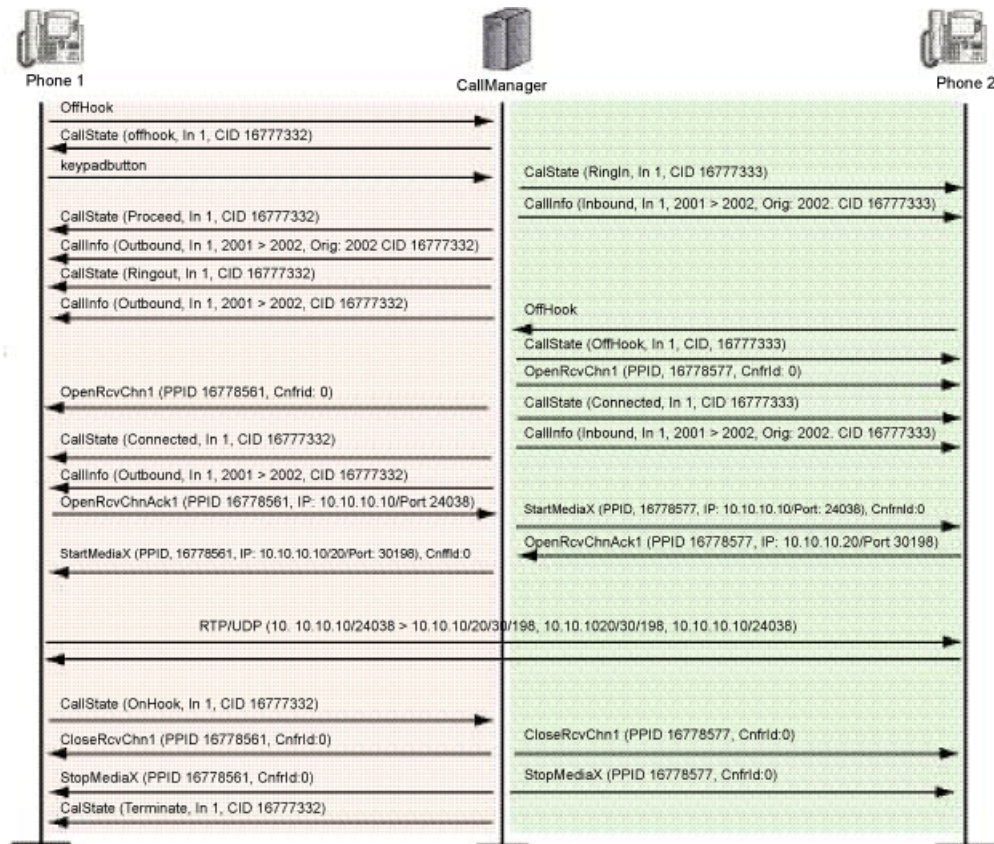
Media Setup

The CallManager sends the IP address and port number of the called party to the calling party. The CallManager also sends the media IP address and port number of the calling party to the called party. After media setup, media is transmitted directly between clients. When the call ends, the CallManager is informed and terminates the media streams. At no time during this process does the CallManager hand over call-setup function to the client. Media is streamed directly between clients through RTP/UDP/IP.

SCCP Control Messages and RTP Flow

Figure 28 on page 249 shows the SCCP control messages used to set up and tear down a simple call between Phone 1 and Phone 2. Except for the OffHook message initiating the call from Phone1 and the OnHook message signaling the end of the call, all aspects of the call are controlled by the CallManager.

Figure 28: Call Setup and Teardown



SCCP Messages

Table 31 on page 249, Table 32 on page 249, Table 33 on page 250, and Table 34 on page 250 list the SCCP call message IDs in the four intervals allowed by the device.

Table 31: Station to CallManager Messages

#define STATION_REGISTER_MESSAGE	0x00000001
#define STATION_IP_PORT_MESSAGE	0x00000002
#define STATION_ALARM_MESSAGE	0x00000020
#define STATION_OPEN_RECEIVE_CHANNEL_ACK	0x00000022

Table 32: CallManager to Station Messages

#define STATION_START_MEDIA_TRANSMISSION	0x00000001
#define STATION_STOP_MEDIA_TRANSMISSION	0x00000002
#define STATION_CALL_INFO_MESSAGE	0x00000020

Table 32: CallManager to Station Messages (*continued*)

#define STATION_OPEN_RECEIVE_CHANNEL_ACK	0x00000022
#define STATION_CLOSE_RECEIVE_CHANNEL	0x00000106

Table 33: CallManager 4.0 Messages and Post Sccp 6.2

#define STATION_REGISTER_TOKEN_REQ_MESSAGE	0x00000029
#define STATION_MEDIA_TRANSMISSION_FAILURE	0x0000002A
#define STATION_OPEN_MULTIMEDIA_RECEIVE_CHANNEL_ACK	0x00000031

Table 34: CallManager to Station

#define STATION_OPEN_MULTIMEDIA_RECEIVE_CHANNEL	0x00000131
#define STATION_START_MULTIMEDIA_TRANSMISSION	0x00000132
#define STATION_STOP_MULTIMEDIA_TRANSMISSION	0x00000133
#define STATION_CLOSE_MULTIMEDIA_RECEIVE_CHANNEL	0x00000136

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - ALG Overview on page 169
 - SCCP ALG Configuration Overview on page 250
 - Example: Configuring the SCCP ALG CallManager/TFTP Server in the Private Zone (CLI) on page 256

SCCP ALG Configuration Overview

The Skinny Client Control Protocol Application Layer Gateway (SCCP ALG) is enabled by default on the device—no action is required to enable it. However, you might choose to fine-tune SCCP ALG operations by using the following instructions:

1. Conserve network resources and maximize throughput. For instructions, see “Example: Setting SCCP ALG Inactive Media Timeouts” on page 251.
2. Enable unknown messages to pass when the session is in Network Address Translation (NAT) mode and route mode. For instructions, see “Example: Allowing Unknown SCCP ALG Message Types” on page 253.
3. Protect the SCCP clients from denial-of-service (DoS) flood attacks. For instructions, see “Example: Configuring SCCP ALG DoS Attack Protection” on page 255.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Understanding SCCP ALGs on page 245
- Example: Configuring the SCCP ALG CallManager/TFTP Server in the Private Zone (CLI) on page 256
- Verifying SCCP ALG Configurations on page 257

SCCP ALG Inactive Media Timeout

- Understanding SCCP ALG Inactive Media Timeouts on page 251
- Example: Setting SCCP ALG Inactive Media Timeouts on page 251

Understanding SCCP ALG Inactive Media Timeouts

The inactive media timeout feature helps you to conserve network resources and maximize throughput.

This parameter indicates the maximum length of time (in seconds) a call can remain active without any media traffic within a group. Each time a Real-Time Transport Protocol (RTP) or Real-Time Control Protocol (RTCP) packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the gates the Skinny Client Control Protocol (SCCP) opened for media are closed. The default setting is 120 seconds, and the range is from 10 to 2550 seconds. Note that upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SCCP ALGs on page 245
 - SCCP ALG Configuration Overview on page 250
 - Example: Setting SCCP ALG Inactive Media Timeouts on page 251

Example: Setting SCCP ALG Inactive Media Timeouts

This example shows how to set the inactive media timeout value for the SCCP ALG.

- Requirements on page 251
- Overview on page 251
- Configuration on page 252
- Verification on page 252

Requirements

Before you begin, review the parameter used to indicate the maximum length of time (in seconds) a call can remain active without any media traffic within a group. See “Understanding SCCP ALG Inactive Media Timeouts” on page 251.

Overview

Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the gates the SCCP opened for media are closed. This example sets the media inactivity timeout to 90 seconds.

Configuration

J-Web Quick Configuration

To set the inactive media timeout for the SCCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **SCCP** tab.
3. In the Inactive Media Timeout box, enter **90**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To set the inactive media timeout for the SCCP ALG:

1. Configure the SCCP ALG inactive media timeout value.

[edit]
user@host# **set security alg sccp inactive-media-timeout 90**
2. If you are done configuring the device, commit the configuration.

[edit]
user@host# **commit**

Verification

To verify the configuration is working properly, enter the **show security alg sccp** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding SCCP ALG Inactive Media Timeouts on page 251
- SCCP ALG Configuration Overview on page 250
- Verifying SCCP ALG Configurations on page 257

SCCP ALG Unknown Message Types

- Understanding SCCP ALG Unknown Message Types on page 252
- Example: Allowing Unknown SCCP ALG Message Types on page 253

Understanding SCCP ALG Unknown Message Types

To accommodate on-going development of the Skinny Client Control Protocol (SCCP), you might want to allow traffic containing new SCCP message types. The unknown SCCP message type feature enables you to configure the device to accept SCCP traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.

This feature enables you to specify how unidentified SCCP messages are handled by the device. The default is to drop unknown (unsupported) messages. We do not recommend permitting unknown messages because they can compromise security. However, in a secure test or production environment, this command can be useful for resolving

interoperability issues with disparate vendor equipment. Permitting unknown SCCP messages can help you get your network operational so that you can later analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped.

Note that this command applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SCCP ALGs on page 245
 - SCCP ALG Configuration Overview on page 250
 - Example: Allowing Unknown SCCP ALG Message Types on page 253

Example: Allowing Unknown SCCP ALG Message Types

This example shows how to configure the SCCP ALG to allow unknown SCCP message types in both NAT mode and route mode.

- Requirements on page 253
- Overview on page 253
- Configuration on page 253
- Verification on page 254

Requirements

Before you begin, determine whether to accommodate new and unknown SCCP message types for the device. See "Understanding SCCP ALG Unknown Message Types" on page 252.

Overview

This feature enables you to specify how unidentified SCCP messages are handled by a Juniper Networks device. The default is to drop unknown (unsupported) messages because unknown messages can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment.

Configuration

J-Web Quick Configuration

To configure the SCCP ALG to allow unknown message types:

1. Select **Configure>Security>ALG**.
2. Select the **SCCP** tab.
3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

**Step-by-Step
Procedure**

To configure the SCCP ALG to allow unknown message types:

1. Allow unknown message types to pass if the session is in either NAT mode or in route mode.

```
[edit]
user@host# set security alg sccp application-screen unknown-message
permit-nat-applied permit-routed
```

2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg sccp** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding SCCP ALG Unknown Message Types on page 252
- SCCP ALG Configuration Overview on page 250
- Verifying SCCP ALG Configurations on page 257

SCCP ALG DoS Attack Protection

- Understanding SCCP ALG DoS Attack Protection on page 254
- Example: Configuring SCCP ALG DoS Attack Protection on page 255

Understanding SCCP ALG DoS Attack Protection

You can protect Skinny Client Control Protocol Application Layer Gateway (SCCP ALG) clients from denial-of-service (DoS) flood attacks by limiting the number of calls they attempt to process.

When you configure SCCP call flood protection, the SCCP ALG drops any calls exceeding the threshold you set. The range is 2 to 1000 calls per second per client, the default is 20.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding SCCP ALGs on page 245
- SCCP ALG Configuration Overview on page 250
- Example: Configuring SCCP ALG DoS Attack Protection on page 255

Example: Configuring SCCP ALG DoS Attack Protection

This example shows how to configure connection flood protection for the SCCP ALG.

- Requirements on page 255
- Overview on page 255
- Configuration on page 255
- Verification on page 255

Requirements

Before you begin, determine whether to protect the SCCP media gateway from DoS flood attacks. See “Understanding SCCP ALG DoS Attack Protection” on page 254.

Overview

In this example, the device is configured to drop any calls exceeding 500 per second per client.

Configuration

J-Web Quick Configuration

To configure call flood protection for the SCCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **SCCP** tab.
3. In the Call flood threshold box, type **500**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure call flood protection for the SCCP ALG:

1. Configure the DoS attack protection:


```
[edit]
user@host# set security alg sccp application-screen call-flood threshold 500
```
2. If you are done configuring the device, commit the configuration.


```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg sccp** command.

Related Topics

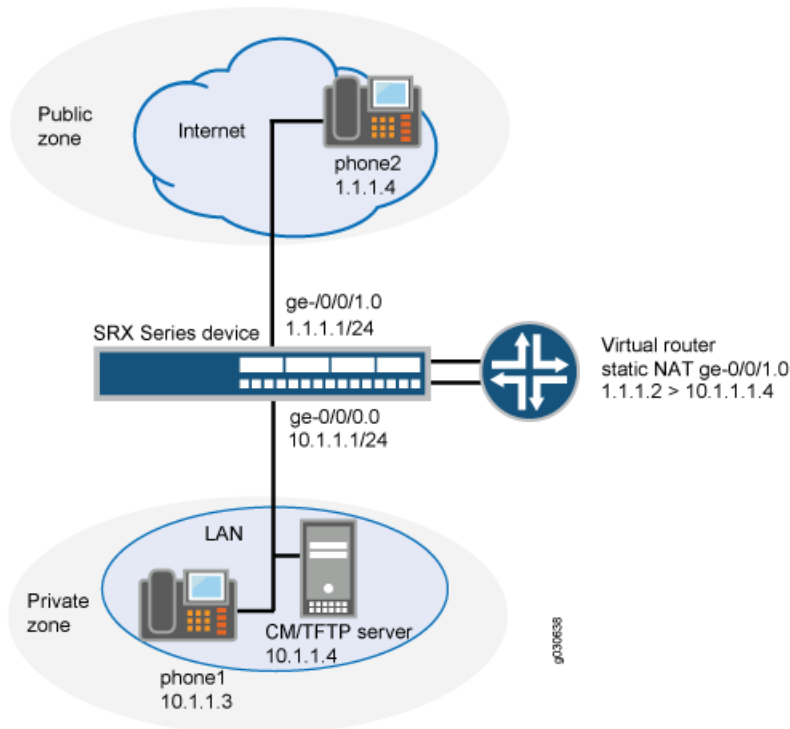
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding SCCP ALG DoS Attack Protection on page 254
- SCCP ALG Configuration Overview on page 250
- Verifying SCCP ALG Configurations on page 257

Example: Configuring the SCCP CallManager/TFTP Server in the Private Zone (CLI)

When the same device serves as both the CallManager and the TFTP server and are located in the private network, you might want to configure static NAT on the outgoing interface of the Juniper Networks device.

In this example, phone1 and the CallManager/TFTP server are on the ge-0/0/0.0 interface in the private zone, and phone2 is on the ge-0/0/1.0 interface in the public zone. You configure static NAT for the CallManager/TFTP server on the ge-0/0/1.0 interface, so that when phone2 boots up it can contact the TFTP server and obtain the IP address of the CallManager. (We recommend that you change the IP address of the CallManager in the TFTP server config file (sep <mac_addr>.cnf) to the NAT IP address of the CallManager.) You then create a policy allowing SCCP traffic from the public to the private zone and reference that NAT in the policy. You also create a policy from the Trust to the Untrust zone to allow phone1 to call out. See Figure 29 on page 256.

Figure 29: CallManager/TFTP Server in the Private Zone



To configure the SCCP CallManager/TFTP server in the private zone:

1. Configure interfaces.

```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 1.1.1.1/24
```
2. Configure the zone.

```
user@host# set security zones security-zone private interface ge-0/0/0.0
```



```

user@host# set security zones security-zone private address-book address phone1
10.1.1.3/32
user@host# set security zones security-zone private address-book address
cm-tftp_server 10.1.1.4/32
user@host# set security zones security-zone public interface ge-0/0/1.0
user@host# set security zones security-zone public address-book address phone2
1.1.1.4/32

```

3. Configure static NAT.

```
user@host# set security nat interface ge-0/0/1.0 static 1.1.1.2 host 10.1.1.4
```

4. Configure policies.

```

user@host# set security policies from-zone private to-zone public policy out-pol
match source-address any
user@host# set security policies from-zone private to-zone public policy out-pol
match destination-address phone2
user@host# set security policies from-zone private to-zone public policy out-pol
match application junos-sccp
user@host# set security policies from-zone private to-zone public policy out-pol then
permit source-nat interface
user@host# set security policies from-zone public to-zone junos-global policy in-pol
match source-address phone2
user@host# set security policies from-zone public to-zone junos-global policy in-pol
match destination-address static_nat_1.1.1.2_32
user@host# set security policies from-zone public to-zone junos-global policy in-pol
match application junos-sccp
user@host# set security policies from-zone public to-zone junos-global policy in-pol
then permit

```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding SCCP ALGs on page 245
 - SCCP ALG Configuration Overview on page 250
 - Verifying SCCP ALG Configurations on page 257

Verifying SCCP ALG Configurations

- Verifying SCCP ALGs on page 257
- Verifying SCCP Calls on page 258
- Verifying SCCP Call Details on page 258
- Verifying SCCP Counters on page 259

Verifying SCCP ALGs

Purpose Display SCCP verification options.

Action From the CLI, enter the **show security alg sccp** command.

```
user@host> show security alg sccp ?
```

Possible completions:

calls	Show SCCP calls
counters	Show SCCP counters

Meaning The output shows a list of all SCCP verification parameters. Verify the following information:

- All SCCP calls
- Counters for all SCCP calls

Verifying SCCP Calls

Purpose Display a list of all SCCP calls

Action From the CLI, enter the **show security alg sccp calls** command.

```
user@host> show security alg sccp calls
```

Possible completions:

calls	Show SCCP calls
counters	Show SCCP counters
endpoints	Show SCCP endpoints

Meaning The output shows a list of all SCCP verification parameters. Verify the following information:

- All SCCP calls
- Counters for all SCCP calls
- Information about all SCCP endpoints

Verifying SCCP Call Details

Purpose Display details about all SCCP calls.

Action From the CLI, enter the **show security alg sccp calls detail** command.

```
user@host> show security alg sccp calls detail
```

```
Client IP address: 11.0.102.91
```

```
Client zone: 7
```

```
CallManager IP: 13.0.99.226
```

```
Conference ID: 16789504
```

```
Resource manager group: 2048
```

```
SCCP channel information:
```

```
Media transmit channel address (IP address/Port): 0.0.0.0:0
```

```
Media transmit channel translated address (IP address/Port): 0.0.0.0:0
```

```
Media transmit channel pass-through party ID (PPID): 0
```

```
Media transmit channel resource ID: 0
```

```
Media receive channel address (IP address/Port): 11.0.102.91:20060
```

```
Media receive channel translated address (IP address/Port): 25.0.0.1:1032
```

```
Media receive channel pass-through party ID (PPID): 16934451
```

```
Media receive channel resource ID: 8185
```

```
Multimedia transmit channel address (IP address/Port): 0.0.0.0:0
```

```
Multimedia transmit channel translated address (IP address/Port): 0.0.0.0:0
```

```
Multimedia transmit channel pass-through party ID (PPID): 0
```

```
Multimedia transmit channel resource ID: 0
```

```
Multimedia receive channel address (IP address/Port): 0.0.0.0:0
```

```
Multimedia receive channel translated address (IP address/Port): 0.0.0.0:0
```

```

Multimedia receive channel pass-through party ID (PPID): 0
Multimedia receive channel resource ID: 0
Total number of calls = 1

```

Meaning The output shows a list of all SCCP verification parameters. Verify the following information:

- Client zone
- CallManager IP address: 13.0.99.226
- Conference ID
- Resource manager group
- SCCP channel information
- Total number of calls

Verifying SCCP Counters

Purpose Display a list of all SCCP counters

Action From the J-Web interface, select **Monitor>ALGs>SCCP>Counters**. Alternatively, from the CLI, enter the **show security alg sccp counters** command.

```

user@host> show security alg sccp counters
SCCP call statistics:
  Active client sessions      : 0
  Active calls                : 0
  Total calls                 : 0
  Packets received           : 0
  PDUs processed              : 0
  Current call rate          : 0
Error counters:
  Packets dropped             : 0
  Decode errors               : 0
  Protocol errors             : 0
  Address translation errors  : 0
  Policy lookup errors        : 0
  Unknown PDUs                : 0
  Maximum calls exceeded      : 0
  Maximum call rate exceeded  : 0
  Initialization errors       : 0
  Internal errors              : 0
  Nonspecific error           : 0
  No active calls to delete   : 0
  No active client sessions to delete : 0
  Session cookie create errors : 0
  Invalid NAT cookie detected  : 0

```

Meaning The output shows a list of all SCCP verification parameters. Verify the following information:

- SCCP call statistics
- Error counters

Related Topics • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- [SCCP ALG Configuration Overview](#) on page 250
- [Example: Configuring the SCCP ALG CallManager/TFTP Server in the Private Zone \(CLI\)](#) on page 256

CHAPTER 14

MGCP ALGs

- Understanding MGCP ALGs on page 261
- MGCP ALG Configuration Overview on page 267
- MGCP ALG Call Duration and Timeouts on page 267
- MGCP ALG DoS Attack Protection on page 272
- MGCP ALG Unknown Message Types on page 274
- Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs on page 275
- Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALGs and NAT (CLI) on page 282

Understanding MGCP ALGs

The Media Gateway Control Protocol (MGCP) is a text-based Application Layer protocol used for call setup and call control between the media gateway and the media gateway controller (MGC).

The protocol is based on a master/slave call control architecture: the MGC (call agent) maintains call control intelligence, and media gateways carry out the instructions from the call agent. Both signaling packets and media packets are transmitted over UDP. Junos OS supports MGCP in route mode and Network Address Translation (NAT) mode.

The MGCP Application Layer Gateway (ALG) performs the following procedures:

- Conducts voice-over-IP (VoIP) signaling payload inspection. The payload of the incoming VoIP signaling packet is fully inspected based on related RFCs and proprietary standards. Any malformed packet attack is blocked by the ALG.
- Conducts MGCP signaling payload inspection. The payload of the incoming MGCP signaling packet is fully inspected in accordance with RFC 3435. Any malformed-packet attack is blocked by the ALG.
- Provides stateful processing. The corresponding VoIP-based state machines are invoked to process the parsed information. Any out-of-state or out-of-transaction packet is identified and properly handled.

- Performs NAT. Any embedded IP address and port information in the payload is properly translated based on the existing routing information and network topology, and is then replaced with the translated IP address and port number, if necessary.
- Manages pinholes for VoIP traffic. To keep the VoIP network secure, the IP address and port information used for media or signaling is identified by the ALG, and any needed pinhole is dynamically created and closed during call setup.

This topic contains the following sections:

- MGCP Security on page 262
- Entities in MGCP on page 262
- Commands on page 264
- Response Codes on page 266

MGCP Security

The MGCP ALG includes the following security features:

- Denial-of-service (DoS) attack protection. The ALG performs stateful inspection at the UDP packet level, the transaction level, and the call level. MGCP packets matching the RFC 3435 message format, transaction state, and call state, are processed. All other messages are dropped.
- Security policy enforcement between gateway and gateway controller (signaling policy).
- Security policy enforcement between gateways (media policy).
- Per-gateway MGCP message flooding control. Any malfunctioning or hacked gateway will not disrupt the whole VoIP network. Combined with per-gateway flooding control, damage is contained within the impacted gateway.
- Per-gateway MGCP connection flooding control.
- Seamless switchover/failover if calls, including calls in progress, are switched to the standby firewall in case of system failure.

Entities in MGCP

There are four basic entities in MGCP:

- Endpoint on page 262
- Connection on page 263
- Call on page 263
- Call Agent on page 263

Endpoint

A media gateway is a collection of endpoints. An endpoint can be an analog line, trunk, or any other access point. An endpoint contains the following elements:

`local-endpoint-name@domain-name`

The following examples are some valid endpoint IDs:

```
group1/Trk8@mynetwork.net
group2/Trk1/*@[192.168.10.8] (wild-carding)
$@voiptel.net (any endpoint within the media gateway)
*@voiptel.net (all endpoints within the media gateway)
```

Connection

Connections are created on each endpoint by an MG during call setup. A typical VoIP call involves two connections. A complex call, for example a three-party call or conference call, might require more connections. The MGC can instruct media gateways to create, modify, delete, and audit a connection.

A connection is identified by its connection ID, which is created by the MG when it is requested to create a connection. Connection ID is presented as a hexadecimal string, and its maximum length is 32 characters.

Call

A call is identified by its call ID, which is created by the MGC when establishing a new call. Call ID is a hexadecimal string with a maximum length of 32 characters. Call ID is unique within the MGC. Two or more connections can have the same call ID if they belong to the same call.

Call Agent

One or more call agents (also called media gateway controllers) are supported in MGCP to enhance reliability in the VoIP network. The following two examples are of call agent names:

```
CallAgent@voipCA.mynetwork.com
voipCA.mynetwork.com
```

Several network addresses can be associated under one domain name in the Domain Name System (DNS). By keeping track of the time to live (TTL) of DNS query/response data and implementing retransmission using other alternative network addresses, switchover and failover is achieved in MGCP.

The concept of a *notified entity* is essential in MGCP. The notified entity for an endpoint is the call agent currently controlling that endpoint. An endpoint should send any MGCP command to its notified entity. However, different call agents might send MGCP commands to this endpoint.

The notified entity is set to a provisioned value upon startup, but can be changed by a call agent through the use of the **NotifiedEntity** parameter contained in an MGCP message. If the notified entity for an endpoint is empty or has not been set explicitly, its value defaults to the source address of the last successful non-audit MGCP command received for that endpoint.

Commands

The MGCP protocol defines nine commands for controlling endpoints and connections. All commands are composed of a command header, optionally followed by Session Description Protocol (SDP) information. A command header has the following elements:

- A command line: command verb + transaction ID + endpointId + MGCP version.
- Zero or more parameter lines, composed of a parameter name followed by a parameter value.

Table 35 on page 264 lists supported MGCP commands and includes a description of each, the command syntax, and examples. Refer to RFC 2234 for a complete explanation of command syntax.

Table 35: MGCP Commands

Command	Description	Command Syntax	Example
EPCF	EndpointConfiguration—Used by a call agent to inform a gateway of coding characteristics (a-law or mu-law) expected by the line side of the endpoint.	ReturnCode [PackageList] EndpointConfiguration (EndpointId,[BearerInformation])	EPCF 2012 wxx/T2@mynet.com MGCP 1.0B: e:mu
CRCX	CreateConnection—Used by a call agent to instruct the gateway to create a connection with, and endpoint inside, the gateway.	ReturnCode, [ConnectionId,] [SpecificEndPointId,] [LocalConnectionDescriptor,] [SecondEndPointId,] [SecondConnectionId,] [PackageList] CreateConnection (CallId, EndpointId, [NotifiedEntity,] [LocalConnectionOption,] Mode, [{RemoteConnectionDescriptor SecondEndPointId},] [encapsulated RQNT,] [encapsulated EPCF])	CRCX 1205 aaln/1@gw-25.att.net MGCP 1.0C: A3C47F21456789F0L: p:10, a:PCMUM: sendrecvX: 0123456789ADR: L/hdS: L/rgv=0o=- 25678 753849 IN IP4 128.96.41.1s=-c=IN IP4 128.96.41.1t=0 Om=audio 3456 RTP/AVP 0
MDCX	ModifyConnection—Used by a call agent to instruct a gateway to change the parameters for an existing connection.	ReturnCode, [LocalConnectionDescriptor,] [PackageList] ModifyConnection (CallId, EndpointId, ConnectionId, [NotifiedEntity,] [LocalConnectionOption,] [Mode,] [RemoteConnectionDescriptor,] [encapsulated RQNT,] [encapsulated EPCF])	MDCX 1210 aaln/1@rgw-25.att.net MGCP 1.0C: A3C47F21456789F0I: FDE234C8M: recvonlyX: 0123456789AER: L/huS: G/rtv=0o=- 4723891 7428910 IN IP4 128.96.63.25s=-c=IN IP4 128.96.63.25t=0 Om=audio 3456 RTP/AVP 0

Table 35: MGCP Commands (*continued*)

Command	Description	Command Syntax	Example
DLCX	<p>DeleteConnection—Used by a call agent to instruct a gateway to delete an existing connection.</p> <p>DeleteConnection can also be used by a gateway to release a connection that can no longer be sustained.</p>	ReturnCode, ConnectionParameters, [PackageList] DeleteConnection (CallId, EndpointId, ConnectionId, [NotifiedEntity,] [encapsulated RQNT,] [encapsulated EPCF])	<p>Example 1: MGC -> MG</p> <p>DLCX 9210 aaln/1@rgw-25.att.net MGCP 1.0C: A3C47F21456789F0I: FDE234C8</p> <p>Example 2: MG -> MGC</p> <p>DLCX 9310 aaln/1@rgw-25.att.net MGCP 1.0C: A3C47F21456789F0I: FDE234C8E: 900 - Hardware errorP: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48</p>
RQNT	NotificationRequest command—Used by a call agent to instruct an MG to monitor for certain event(s) or signal(s) for a specific endpoint.	ReturnCode, [PackageList] NotificationRequest([EndpointId, [NotifiedEntity,] [RequestedEvents,] RequestIdentifier, [DigitMap,] [SignalRequests,] [QuarantineHandling,] [DetectEvents,] [encapsulated EPCF])	<p>RQNT 1205 aaln/1@rgw-25.att.net MGCP 1.0N: ca-new@callagent-ca.att.netX: 0123456789AAR: L/hd(A, E(S(L/d),R(L/oc,L/hu,D/[0-9#*T](D))))D: (0T 00T xx 9 xxxxxxxxxx 90 1x.T)S:T: G/ft</p>
NTFY	Notify—Used by a gateway to inform the call agent when requested event(s) or signal(s) occur.	ReturnCode, [PackageList] Notify (EndpointId, [NotifiedEntity,] RequestIdentifier, ObservedEvents)	<p>NTFY 2002 aaln/1@rgw-25.att.net MGCP 1.0N: ca@ca1.att.net:5678X: 0123456789ACO: L/hd,D/9,D/1,D/2,D/0,D/1,D/8,D/2,D/9,D/4, D/2,D/6,D/6</p>
AUEP	AuditEndpoint—Used by a call agent to audit the status of the endpoint.	ReturnCode, EndPointIdList, { [RequestedEvents,] [QuarantineHandling,] [DigitMap,] [SignalRequests,] [RequestedIdentifier,] [NotifiedEntity,] [ConnectionIdentifier,] [DetectEvents,] [ObservedEvents,] [EventStats,] [BearerInformation,] [BearerMethod,] [RestartDelay,] [ReasonCode,] [MaxMGCPDatagram,] [Capabilities]} [PackageList] AuditEndpoint (EndpointId, [RequestedInfo])	<p>Example 1:</p> <p>AUEP 1201 aaln/1@rgw-25.att.net MGCP 1.0F: A, R,D,S,X,N,I,T,O</p> <p>Example 2:</p> <p>AUEP 1200 *@rgw-25.att.net MGCP 1.0</p>
AUCX	AuditConnection—Used by a call agent to collect the parameters applied to a connection.	ReturnCode, [CallId,] [NotifiedEntity,] [LocalConnectionOptions,] [Mode,] [RemoteConnectionDescriptor,] [LocalConnectionDescriptor,] [ConnectionParameters,] [PackageList] AuditConnection (EndpointId, ConnectionId, RequestedInfo)	<p>AUCX 3003 aaln/1@rgw-25.att.net MGCP 1.0I: 32F345E2F: C,N,L,M,LC,P</p>
RSIP	RestareInProgress—Used by a gateway to notify a call agent that one or more endpoints are being taken out of service or placed back in service.	ReturnCode, [NotifiedEntity,] [PackageList] RestartInProgress (EndpointId, RestartMethod, [RestartDelay,] [ReasonCode])	<p>RSIP 5200 aaln/1@rg2-25.att.net MGCP 1.0RM: gracefulRD: 300</p>

Response Codes

Every command sent by the calling agent or gateway, whether successful or not, requires a response code. The response code is in the header of the response message, and optionally is followed by session description information.

The response header is composed of a response line, followed by zero or more parameter lines, each containing a parameter name letter followed by its value. The response header is composed of a three-digit response code, transaction ID, and optionally followed by commentary. The response header in the following response message shows response code 200 (successful completion), followed by ID 1204 and the comment:OK.

```
200 1204 OK
I: FDE234C8
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 96
a=rtpmap:96 G726-32/8000
```

The ranges of response codes are defined as follows:

- 000 — 099 indicate a response acknowledgement.
- 100 — 199—indicate a provisional response.
- 200 — 299 indicate a successful completion (final response).
- 400 — 499 indicate a transient error (final response).
- 500 — 599 indicate a permanent error (final response).

Refer to RFC 3661 for detailed information about response codes.

A response to a command is sent to the source address of the command, not to the current notified entity. A media gateway can receive MGCP commands from various network addresses simultaneously, and send back responses to corresponding network addresses. However, it sends all MGCP commands to its current notified entity.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- ALG Overview on page 169
- MGCP ALG Configuration Overview on page 267
- Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs on page 275
- Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALGs and NAT (CLI) on page 282

MGCP ALG Configuration Overview

The Media Gateway Control Protocol (MGCP ALG) is enabled by default on the device—no action is required to enable it. However, you might choose to fine-tune MGCP ALG operations by using the following instructions:

1. Free up bandwidth when calls fail to properly terminate. See “Example: Setting MGCP ALG Call Duration” on page 268.
2. Control how long a call can remain active without any media traffic. See “Example: Setting MGCP ALG Inactive Media Timeout” on page 270.
3. Track and clear signaling traffic when it times out. See “Example: Setting MGCP ALG Transaction Timeout” on page 271.
4. Protect the media gateway from denial-of-service (DoS) flood attacks. See “Example: Configuring MGCP ALG DoS Attack Protection” on page 272.
5. Enable unknown messages to pass when the session is in Network Address Translation (NAT) mode and route mode. See “Example: Allowing Unknown MGCP ALG Message Types” on page 274.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding MGCP ALGs on page 261
 - Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs on page 275
 - Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALGs and NAT (CLI) on page 282
 - Verifying MGCP ALG Configurations

MGCP ALG Call Duration and Timeouts

- Understanding MGCP ALG Call Duration and Timeouts on page 267
- Example: Setting MGCP ALG Call Duration on page 268
- Example: Setting MGCP ALG Inactive Media Timeout on page 270
- Example: Setting MGCP ALG Transaction Timeout on page 271

Understanding MGCP ALG Call Duration and Timeouts

The call duration feature gives you control over Media Gateway Control Protocol (MGCP) call activity and helps you to manage network resources.

Typically a Delete Connection (DLCX) message will be sent out to delete a connection. The MGCP Application Layer Gateway (ALG) intercepts it and removes all media sessions for that connection.

A call can have one or more voice channels. Each voice channel has two sessions (or two media streams), one for Real-Time Transport Protocol (RTP) traffic and one for Real-Time Control Protocol (RTCP) signaling. When managing the sessions, the device considers the sessions in each voice channel as one group. Timeouts and call duration settings apply to a group as opposed to each session.

The following parameters govern MGCP call activity:

- **maximum-call-duration**—This parameter sets the absolute maximum length of a call. When a call exceeds this parameter setting, the MGCP ALG tears down the call and releases the media sessions. The default setting is 43200 seconds, and the range is from 180 through 432000 seconds. This setting also frees up bandwidth in cases where calls fail to properly terminate.
- **inactive-media-timeout**—This parameter indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the MGCP ALG gates opened for media are closed. The default setting is 120 seconds, and the range is 10 through 2550 seconds. Note that upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated.
- **transaction-timeout**—A transaction is a signaling message, for example, an NTFY from the gateway to the call agent or a 200 OK from the call agent to the gateway. The Juniper Networks device tracks these transactions and clears them when they time out. The timeout range for MGCP transactions is 3 through 50 seconds and the default is 30 seconds.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding MGCP ALGs on page 261
- MGCP ALG Configuration Overview on page 267
- Example: Setting MGCP ALG Call Duration on page 268
- Example: Setting MGCP ALG Inactive Media Timeout on page 270
- Example: Setting MGCP ALG Transaction Timeout on page 271

Example: Setting MGCP ALG Call Duration

This example shows how to set call duration for the MGCP ALG.

- Requirements on page 269
- Overview on page 269
- Configuration on page 269
- Verification on page 269

Requirements

Before you begin, determine the type of parameter used to control the MGCP call activity and manage its network resources. See “Understanding MGCP ALG Call Duration and Timeouts” on page 267.

Overview

The **maximum-call-duration** parameter governs MGCP call activity and sets the absolute maximum length of a call. When a call exceeds this parameter setting, the MGCP ALG tears down the call and releases the media sessions. The default setting is 432000 seconds, and the range is 180 through 432000 seconds. This setting also frees up bandwidth in cases where calls fail to properly terminate. In this example, the call duration is set to 180000 seconds.

Configuration

J-Web Quick Configuration

To set call duration for the MGCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. In the Maximum call duration box, enter **3000**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To set call duration for the MGCP ALG:

1. Configure the MGCP ALG call duration.

```
[edit]
user@host# set security alg mgcp maximum-call-duration 3000
```
2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg mgcp** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding MGCP ALG Call Duration and Timeouts on page 267
- MGCP ALG Configuration Overview on page 267

Example: Setting MGCP ALG Inactive Media Timeout

This example shows how to set the inactive media timeout value for the MGCP ALG.

- Requirements on page 270
- Overview on page 270
- Configuration on page 270
- Verification on page 271

Requirements

Before you begin, determine the type of parameter used to control the MGCP call activity and manage its network resources. See “Understanding MGCP ALG Call Duration and Timeouts” on page 267.

Overview

The **inactive-media-timeout** parameter governs MGCP call activity and indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the MGCP ALG gates opened for media are closed. The default setting is 120 seconds, and the range is from 10 to 2550 seconds. Note that upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated. In this example, the inactive media timeout is set to 90 seconds.

Configuration

J-Web Quick Configuration

To set the inactive media timeout for the MGCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. In the Inactive Media Timeout box, enter **90**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To set the inactive media timeout for the MGCP ALG:

1. Configure the MGCP ALG inactive media timeout value.
[edit]
user@host# **set security alg mgcp inactive-media-timeout 90**
2. If you are done configuring the device, commit the configuration.
[edit]
user@host# **commit**

Verification

To verify the configuration is working properly, enter the **show security alg mgcp** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding MGCP ALG Call Duration and Timeouts on page 267
 - MGCP ALG Configuration Overview on page 267

Example: Setting MGCP ALG Transaction Timeout

This example shows how to set the transaction timeout for the MGCP ALG.

- Requirements on page 271
- Overview on page 271
- Configuration on page 271
- Verification on page 272

Requirements

Before you begin, determine the type of parameter used to control the MGCP call activity and manage its network resources. See “Understanding MGCP ALG Call Duration and Timeouts” on page 267.

Overview

The **transaction-timeout** parameter governs MGCP call activity and is a signaling message; for example, a NOTIFY from the gateway to the call agent or a 200 OK from the call agent to the gateway. The Juniper Networks device tracks these transactions, and clears them when they time out. The timeout range for MGCP transactions is from 3 to 50 seconds, and the default is 30 seconds. In this example, the transaction timeout is set to 20 seconds.

Configuration

J-Web Quick Configuration

To set the transaction timeout for the MGCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. In the Transaction Timeout box, enter **20**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To set the transaction timeout for the MGCP ALG:

1. Configure the MGCP ALG transaction timeout value.

```
[edit]
user@host# set security alg mgcp transaction-timeout 20
```

2. If you are done configuring the device, commit the configuration.

```
[edit]  
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg mgcp** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding MGCP ALG Call Duration and Timeouts on page 267
 - MGCP ALG Configuration Overview on page 267

MGCP ALG DoS Attack Protection

- Understanding MGCP ALG DoS Attack Protection on page 272
- Example: Configuring MGCP ALG DoS Attack Protection on page 272

Understanding MGCP ALG DoS Attack Protection

You can protect the Media Gateway Control Protocol (MGCP) media gateway from denial-of-service (DoS) flood attacks by limiting the number of remote access service (RAS) messages and connections per second it will attempt to process.

When you configure MGCP message flood protection, the MGCP Application Layer Gateway (ALG) drops any messages exceeding the threshold you set. The range is 2 to 50,000 messages per second per media gateway, and the default is 1000 messages per second per media gateway.

When you configure MGCP connection flood protection, the MGCP ALG drops any connection request exceeding the threshold you set. This limits the rate of processing of **CreateConnection (CRCX)** commands, thereby indirectly limiting pinhole creation. The range is 2 to 10,000 connection requests per second per media gateway, the default is 200.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding MGCP ALGs on page 261
 - MGCP ALG Configuration Overview on page 267
 - Example: Configuring MGCP ALG DoS Attack Protection on page 272

Example: Configuring MGCP ALG DoS Attack Protection

This example shows how to configure connection flood protection for the MGCP ALG.

- Requirements on page 273
- Overview on page 273

- Configuration on page 273
- Verification on page 273

Requirements

Before you begin, determine whether to protect the MGCP media gateway from DoS flood attacks. See “Understanding MGCP ALG DoS Attack Protection” on page 272.

Overview

In this example, you configure the MGCP ALG to drop any message requests exceeding 10,000 requests per second and to drop any connection requests exceeding 4000 per second.

Configuration

J-Web Quick Configuration

To configure connection flood protection for the MGCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. In the Message flood gatekeeper threshold box, type **10000**.
4. In the Connection flood threshold box, type **4000**.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure connection flood protection for the MGCP ALG:

1. Configure the connection flood threshold value.

```
[edit]
user@host# set security alg mgcp application-screen message-flood threshold
10000
user@host# set security alg mgcp application-screen connection-flood threshold
4000
```
2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg mgcp** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding MGCP ALG DoS Attack Protection on page 272
- MGCP ALG Configuration Overview on page 267

MGCP ALG Unknown Message Types

- Understanding MGCP ALG Unknown Message Types on page 274
- Example: Allowing Unknown MGCP ALG Message Types on page 274

Understanding MGCP ALG Unknown Message Types

To accommodate on-going development of the Media Gateway Control Protocol (MGCP), you might want to allow traffic containing new MGCP message types. The unknown MGCP message type feature enables you to configure the Juniper Networks device to accept MGCP traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.

This feature enables you to specify how unidentified MGCP messages are handled by the Juniper Networks device. The default is to drop unknown (unsupported) messages. Unknown messages can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment. Permitting unknown MGCP messages can help you get your network operational so that you can later analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped.

Note that this command applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding MGCP ALGs on page 261
 - MGCP ALG Configuration Overview on page 267
 - Example: Allowing Unknown MGCP ALG Message Types on page 274

Example: Allowing Unknown MGCP ALG Message Types

This example shows how to configure the MGCP ALG to allow unknown MGCP message types in both NAT mode and route mode.

- Requirements on page 274
- Overview on page 275
- Configuration on page 275
- Verification on page 275

Requirements

Before you begin, determine whether to accommodate new and unknown MGCP message types for the device. See “Understanding MGCP ALG Unknown Message Types” on page 274.

Overview

This feature enables you to specify how unidentified MGCP messages are handled by a Juniper Networks device. The default is to drop unknown (unsupported) messages, because unknown messages can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment.

Configuration

J-Web Quick Configuration

To configure the MGCP ALG to allow unknown message types:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure the MGCP ALG to allow unknown message types:

1. Allow unknown message types to pass if the session is in either NAT mode or in route mode.

[edit]

```
user@host# set security alg mgcp application-screen unknown-message
permit-nat-applied permit-routed
```

2. If you are done configuring the device, commit the configuration.

[edit]

```
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security alg mgcp** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding MGCP ALG Unknown Message Types on page 274
- MGCP ALG Configuration Overview on page 267

Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs

This example shows how to configure media gateways in subscriber homes using MGCP ALGs.

- Requirements on page 276
- Overview on page 276

- Configuration on page 277
- Verification on page 280

Requirements

Before you begin:

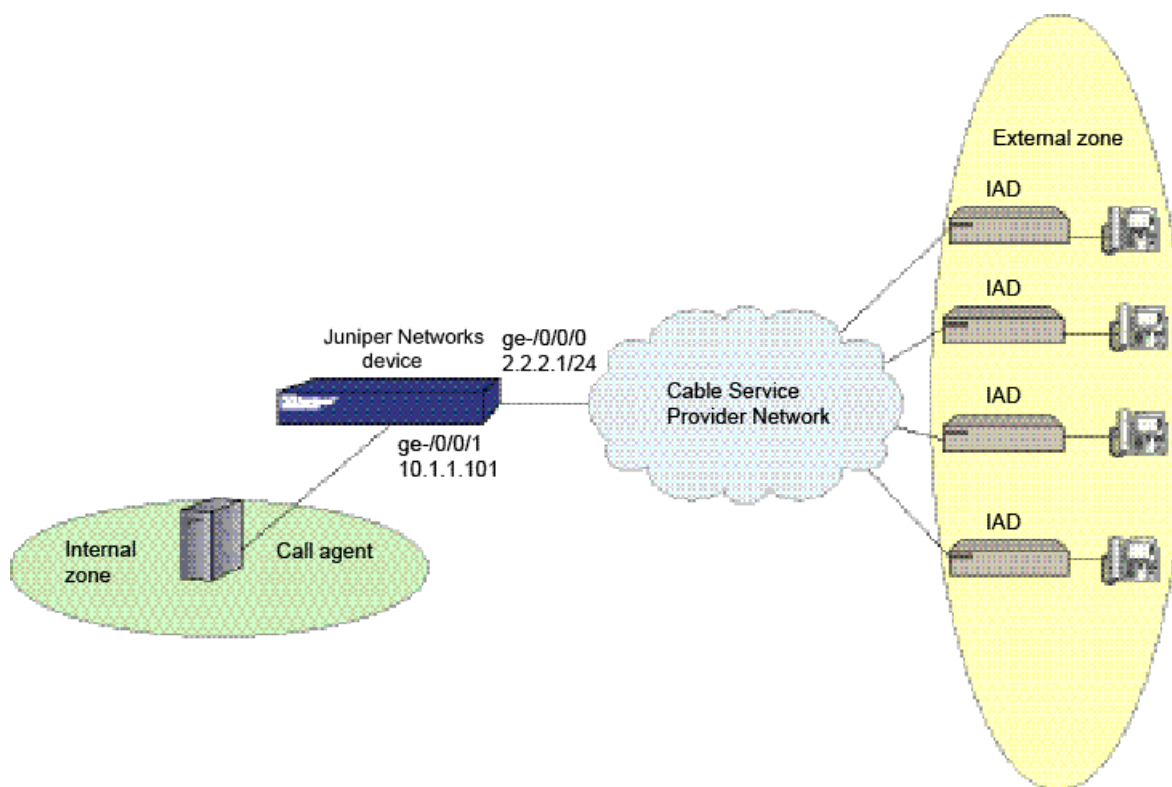
- Configure zones. See “Example: Creating Security Zones” on page 88.
- Configure addresses and interfaces. See “Example: Configuring Address Books” on page 108.
- Configure security policies. See “Security Policies Configuration Overview” on page 120.

Overview

When a cable service provider offers MGCP services to residential subscribers, they locate the Juniper Networks device and call agent on their premises and install a set-top box, in each subscriber's home. The set-top boxes act as gateways for the residences.

After creating zones—`external_subscriber` for the customer and `internal_ca` for the service provider—you configure addresses, then interfaces, and finally policies to allow signaling between endpoints. Note that although gateways frequently reside in different zones, requiring policies for media traffic, in this example both gateways are in the same subnet. Note also that because RTP traffic between the gateways never passes through the device, no policy is needed for the media. See Figure 30 on page 276.

Figure 30: Media Gateway in Subscriber Homes



Configuration

CLI Quick Configuration To quickly configure media gateways in subscriber homes using MGCP ALGs, copy the following commands and paste them into the CLI:

```
[edit]
set security zones security-zone external-subscriber host-inbound-traffic system-services
  all
set security zones security-zone external-subscriber host-inbound-traffic protocols all
set security zones security-zone internal-ca host-inbound-traffic system-services all
set security zones security-zone internal-ca host-inbound-traffic protocols all
set security zones security-zone internal-ca address-book address ca-agent-110.1.1.101/32
set security zones security-zone external-subscriber address-book address
  subscriber-subnet 2.2.2.1/24
set security zones security-zone external-subscriber interfaces ge-0/0/0
set interfaces ge-0/0/0 unit 0 family inet
set security zones security-zone internal-ca interfaces ge-0/0/1
set interfaces ge-0/0/1 unit 0 family inet address 2.2.2.1/24
set security policies from-zone internal-ca to-zone external-subscriber policy
  ca-to-subscribers match source-address ca-agent-1
set security policies from-zone internal-ca to-zone external-subscriber policy
  ca-to-subscribers match destination-address subscriber-subnet
set security policies from-zone internal-ca to-zone external-subscriber policy
  ca-to-subscribers match application junos-mgcp
set security policies from-zone internal-ca to-zone external-subscriber policy
  ca-to-subscribers then permit
set security policies from-zone external-subscriber to-zone internal-ca policy
  subscriber-to-ca match source-address subscriber-subnet
set security policies from-zone external-subscriber to-zone internal-ca policy
  subscriber-to-ca match destination-address ca-agent-1
set security policies from-zone external-subscriber to-zone internal-ca policy
  subscriber-to-ca match application junos-mgcp
set security policies from-zone external-subscriber to-zone internal-ca policy
  subscriber-to-ca then permit
set security policies from-zone internal-ca to-zone internal-ca policy intra-ca match
  source-address any
set security policies from-zone internal-ca to-zone internal-ca policy intra-ca match
  destination-address any
set security policies from-zone internal-ca to-zone internal-ca policy intra-ca match
  application any
set security policies from-zone internal-ca to-zone internal-ca policy intra-ca then permit
set security policies from-zone external-subscriber to-zone external-subscriber policy
  intra-subscriber match source-address any
set security policies from-zone external-subscriber to-zone external-subscriber policy
  intra-subscriber match destination-address any
set security policies from-zone external-subscriber to-zone external-subscriber policy
  intra-subscriber match application any
set security policies from-zone external-subscriber to-zone external-subscriber policy
  intra-subscriber then permit
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure media gateways in subscriber homes using MGCP ALGs:

1. Create security zones for the customer and for the service provider.

```
[edit security zones security-zone external-subscriber]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
[edit security zones security-zone internal-ca]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
```

2. Configure addresses for the zones.

```
[edit]
user@host# set security zones security-zone internal-ca address-book address
ca-agent-1 10.1.1.101/32
user@host# set security zones security-zone external-subscriber address-book
address subscriber-subnet 2.2.2.1/24
```

3. Configure interfaces for the zones.

```
[edit]
user@host# set security zones security-zone external-subscriber interfaces ge-0/0/0
user@host# set interfaces ge-0/0/0 unit 0 family inet
user@host# set security zones security-zone internal-ca interfaces ge-0/0/1
user@host# set interfaces ge-0/0/1 unit 0 family inet address 2.2.2.1/24
```

4. Configure policies for traffic from the internal to the external zone.

```
[edit security policies from-zone internal-ca to-zone external-subscriber policy
ca-to-subscribers]
user@host# set match source-address ca-agent-1
user@host# set match destination-address subscriber-subnet
user@host# set match application junos-mgcp
user@host# set then permit
```

5. Configure policies for traffic from the external to the internal zone.

```
[edit security policies from-zone external-subscriber to-zone internal-ca policy
subscriber-to-ca]
user@host# set match source-address subscriber-subnet
user@host# set match destination-address ca-agent-1
user@host# set match application junos-mgcp
user@host# set then permit
```

6. Configure policies for traffic between two internal zones.

```
[edit security policies from-zone internal-ca to-zone internal-ca policy intra-ca]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set then permit
```

7. Configure policies for traffic between two external zones.

```
[edit security policies from-zone external-subscriber to-zone external-subscriber
policy intra-subscriber]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set then permit
```

Results From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
from-zone internal-ca to-zone external-subscriber {
  policy ca-to-subscribers {
    match {
      source-address ca-agent-1;
      destination-address subscriber-subnet;
      application junos-mgcp;
    }
    then {
      permit;
    }
  }
}
from-zone external-subscriber to-zone internal-ca {
  policy subscriber-to-ca {
    match {
      source-address subscriber-subnet;
      destination-address ca-agent-1;
      application junos-mgcp;
    }
    then {
      permit;
    }
  }
}
from-zone internal-ca to-zone internal-ca {
  policy intra-ca {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone external-subscriber to-zone external-subscriber {
  policy intra-subscriber {
    match {
      source-address any;
      destination-address any;
      application any;
    }
  }
}
```

```

        then {
            permit;
        }
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying MGCP ALGs on page 280
- Verifying MGCP ALG Calls on page 280
- Verifying MGCP ALG Endpoints on page 281
- Verifying MGCP ALG Counters on page 281

Verifying MGCP ALGs

Purpose Verify the MGCP ALG verification options.

Action From operational mode, enter the **show security alg mgcp ?** command.

```

user@host> show security alg mgcp ?
Possible completions:
  calls           Show MGCP calls
  counters        Show MGCP counters
  endpoints       Show MGCP endpoints

```

Meaning The output shows a list of all MGCP verification parameters. Verify the following information:

- All MGCP calls
- Counters for all MGCP calls
- Information about all MGCP endpoints

Verifying MGCP ALG Calls

Purpose Verify information about active MGCP calls.

Action From operational mode, enter the **show security alg mgcp calls** command.

```

user@host> show security alg mgcp calls
Endpoint@GW      Zone      Call ID      RM Group
d001@101.50.10.1  Trust     10d55b81140e0f76  512
  Connection Id> 0
    Local SDP>  o: 101.50.10.1      x_o: 101.50.10.1
                  c: 101.50.10.1/32206    x_c: 101.50.10.1/32206
    Remote SDP> c: 3.3.3.5/16928    x_c: 3.3.3.5/16928
Endpoint@GW      Zone      Call ID      RM Group
d001@3.3.3.5     Untrust   3a104e9b41a7c4c9  511
  Connection Id> 0
    Local SDP>  o: 3.3.3.5      x_o: 3.3.3.5

```



```

c: 3.3.3.5/16928          x_c: 3.3.3.5/16928
Remote SDP> c: 101.50.10.1/32206  x_c: 101.50.10.1/32206

```

Meaning The output displays information about all MGCP calls. Verify the following information:

- Endpoint
- Zone
- Call identifier
- Resource Manager group

Verifying MGCP ALG Endpoints

Purpose Verify information about MGCP endpoints.

Action From operational mode, enter the `show security alg mgcp endpoints` command.

```

user@host> show security alg mgcp endpoints
Gateway: 101.50.10.1 Zone: Trust IP: 101.50.10.1 -> 101.50.10.1
  Endpoint      Trans #  Call #  Notified Entity
  d001          1        1      0.0.0.0/0->0.0.0.0/0
Gateway: 3.3.3.5 Zone: Untrust IP: 3.3.3.5 -> 3.3.3.5
  Endpoint      Trans #  Call #  Notified Entity
  d001          1        1      0.0.0.0/0->0.0.0.0/0

```

Meaning The output displays information about all MGCP endpoints. Verify the following information:

- Gateway IP address and zone of both endpoints
- Endpoint identifier, transaction number, call number, and notified entity for each gateway

Verifying MGCP ALG Counters

Purpose Verify information about MGCP counters.

Action From operational mode, enter the `show security alg mgcp counters` command.

```

user@host> show security alg mgcp counters
MGCP counters summary:
Packets received           :284
Packets dropped            :0
Message received           :284
Number of connections      :4
Number of active connections :3
Number of calls            :4
Number of active calls     :3
Number of transactions     :121
Number of active transactions:52
Number of re-transmission  :68
MGCP Error Counters:
Unknown-method             :0
Decoding error             :0
Transaction error          :0
Call error                 :0
Connection error           :0
Connection flood drop      :0

```

```

Message flood drop      :0
IP resolve error       :0
NAT error               :0
Resource manager error  :0
MGCP Packet Counters:
CRCX      :4      MDCX      :9      DLCX      :2
AUEP      :1      AUCX      :0      NTFY      :43
RSIP      :79     EPCF      :0      RQNT      :51
000-199   :0      200-299   :95     300-999   :0

```

Meaning The output displays information about all MGCP counters. Verify the following information:

- Summary of MGCP counters
- MGCP error counters
- MGCP packet counters

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding MGCP ALGs on page 261
- MGCP ALG Configuration Overview on page 267

Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALGs and NAT (CLI)

When an Internet service provider (ISP) in one geographical location provides service to two networks in different geographical locations, a three-zone configuration might be necessary.

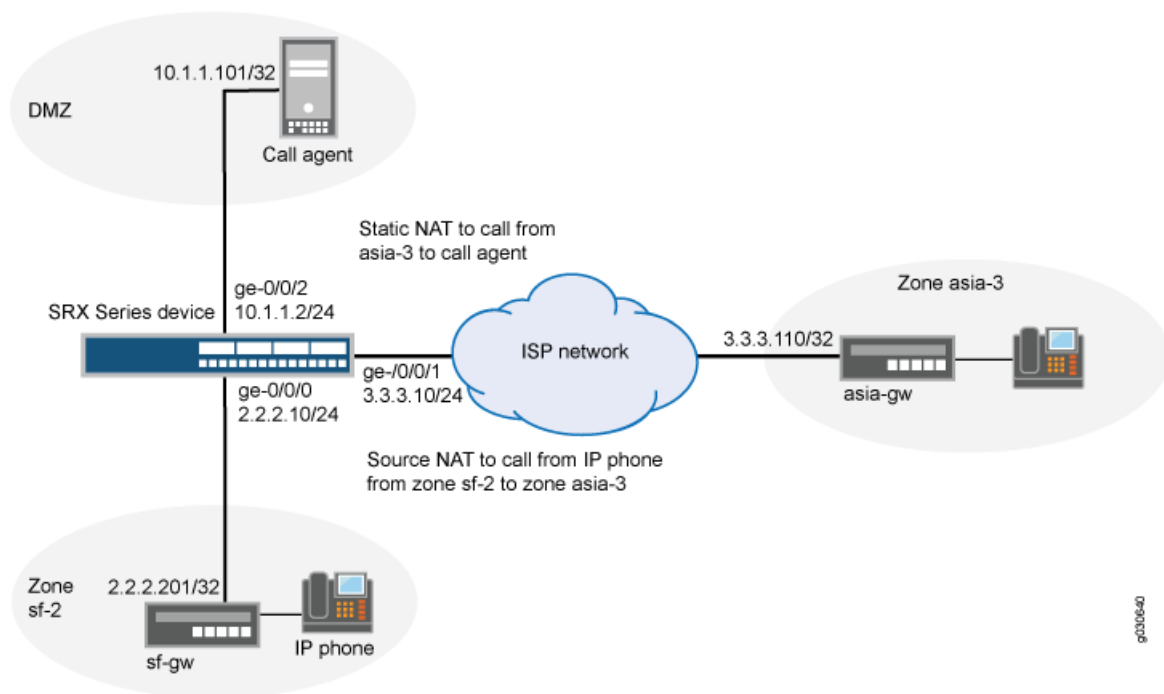
In this example, (see Figure 31 on page 283) an ISP located on the American west coast provides MGCP service to customers in separate networks in Asia and San Francisco. Asia customers are in the asia_3 zone and supported by the asia_gw gateway; San Francisco customers are in the sf_2 zone and supported by the sf_gw gateway; and the west_ca call agent is in the DMZ. The gateways and the call agent are listed in Table 36 on page 282, showing the corresponding IP address, interface, and zone.

Table 36: Three-Zone ISP-Host Service

Gateway	IP Address	Interface	Zone
sf_gw	2.2.2.201	ge-0/0/0	sf_2
asia-gw	3.3.3.110	ge-0/0/1	asia_3
west_ca	10.1.1.101	ge-0/0/2	DMZ

After creating zones and setting addresses for the gateways and the call agent, you associate the zones and addresses to interfaces, and then configure Network Address Translation (NAT) (ge-0/0/1.0) and policies.

Figure 31: Three-Zone ISP-Hosted Service



To configure a three-zone ISP-hosted service using source and static NAT:

1. Configure interfaces.

```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 2.2.2.10/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 3.3.3.10/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 10.1.1.2/24
```

2. Configure addresses.

```
user@host# set security zones security-zone sf_2 address-book address sf_gw
2.2.2.201/32
user@host# set security zones security-zone asia_3 address-book address asia_gw
3.3.3.110/32
user@host# set security zones security-zone dmz address-book address west_ca
10.1.1.101/32
```

3. Associate the zones and addresses to interfaces.

```
user@host# set security zones security-zone sf_2 interfaces ge-0/0/0
user@host# set security zones security-zone asia_3 interfaces ge-0/0/1
user@host# set security zones security-zone dmz interfaces ge-0/0/2
```

4. Configure zones sf_2, asia_3, and DMZ to allow incoming VoIP traffic.

```
user@host# set security zones security-zone sf_2
user@host# set security zones security-zone sf_2 host-inbound-traffic system-services
all
user@host# set security zones security-zone sf_2 host-inbound-traffic protocols all
user@host# set security zones security-zone asia_3
user@host# set security zones security-zone asia_3 host-inbound-traffic
system-services all
```

```
user@host# set security zones security-zone asia_3 host-inbound-traffic protocols
all
user@host# set security zones security-zone dmz
user@host# set security zones security-zone dmz host-inbound-traffic system-services
all
user@host# set security zones security-zone dmz host-inbound-traffic protocols all
```

5. Configure static NAT on interface ge-0/0/ and source NAT on interface ge-0/0/2.

```
user@host# set security nat interface ge-0/0/1.0 static-nat 3.3.3.101/32 host
10.1.1.101/32
user@host# set security nat interface ge-0/0/1.0 source-nat pool src-nat-pool address
2.2.2.10
```

6. Configure policies.

```
user@host# set security policies from-zone dmz to-zone asia_3 policy
pol-dmz-to-asia_3 match source-address west_ca
user@host# set security policies from-zone dmz to-zone asia_3 policy
pol-dmz-to-asia_3 match destination-address asia_gw
user@host# set security policies from-zone dmz to-zone asia_3 policy
pol-dmz-to-asia_3 match application junos-mgcp
user@host# set security policies from-zone dmz to-zone asia_3 policy
pol-dmz-to-asia_3 then permit
user@host# set security policies from-zone asia_3 to-zone dmz policy
pol-asia_3-to-dmz match source-address asia_gw
user@host# set security policies from-zone asia_3 to-zone dmz policy
pol-asia_3-to-dmz match destination-address 3.3.3.101
user@host# set security policies from-zone asia_3 to-zone dmz policy
pol-asia_3-to-dmz match application junos-mgcp
user@host# set security policies from-zone asia_3 to-zone dmz policy
pol-asia_3-to-dmz then permit
user@host# set security policies from-zone sf_2 to-zone dmz policy pol-sf_2-to-dmz
match source-address sf_gw
user@host# set security policies from-zone sf_2 to-zone dmz policy pol-sf_2-to-dmz
match destination-address west-ca
user@host# set security policies from-zone sf_2 to-zone dmz policy pol-sf_2-to-dmz
match application junos-mgcp
user@host# set security policies from-zone sf_2 to-zone dmz policy pol-sf_2-to-dmz
then permit
user@host# set security policies from-zone dmz to-zone sf_2 policy pol-dmz-to-sf_2
match source-address west_ca
user@host# set security policies from-zone dmz to-zone sf_2 policy pol-dmz-to-sf_2
match destination-address sf_gw
user@host# set security policies from-zone dmz to-zone sf_2 policy pol-dmz-to-sf_2
match application junos-mgcp
user@host# set security policies from-zone dmz to-zone sf_2 policy pol-dmz-to-sf_2
then permit
user@host# set security policies from-zone sf_2 to-zone asia_3 policy
pol-sf_2-to-asia_3 match source-address sf_gw
user@host# set security policies from-zone sf_2 to-zone asia_3 policy
pol-sf_2-to-asia_3 match destination-address asia_gw
user@host# set security policies from-zone sf_2 to-zone asia_3 policy
pol-sf_2-to-asia_3 match application junos-mgcp
user@host# set security policies from-zone sf_2 to-zone asia_3 policy
pol-sf_2-to-asia_3 then permit source-nat pool src-nat-pool
```

```
user@host# set security policies from-zone sf_2 to-zone asia_3 policy
pol-sf_2-to-asia_3 match source-address sf_gw
user@host# set security policies from-zone sf_2 to-zone asia_3 policy
pol-sf_2-to-asia_3 match destination-address asia_gw
user@host# set security policies from-zone sf_2 to-zone asia_3 policy
pol-sf_2-to-asia_3 match application junos-mgcp
user@host# set security policies from-zone sf_2 to-zone asia_3 policy
pol-sf_2-to-asia_3 then permit source-nat pool src-nat-pool
user@host# set security policies from-zone sf_2 to-zone asia_3 policy pol-intra-sf_2
match source-address any
user@host# set security policies from-zone sf_2 to-zone asia_3 policy pol-intra-sf_2
match destination-address any
user@host# set security policies from-zone sf_2 to-zone asia_3 policy pol-intra-sf_2
match application any
user@host# set security policies from-zone sf_2 to-zone asia_3 policy pol-intra-sf_2
then permit
user@host# set security policies from-zone sf_2 to-zone asia_3 policy pol-intra-asia_3
match source-address any
user@host# set security policies from-zone sf_2 to-zone asia_3 policy pol-intra-asia_3
match destination-address any
user@host# set security policies from-zone sf_2 to-zone asia_3 policy pol-intra-asia_3
match application any
user@host# set security policies from-zone sf_2 to-zone asia_3 policy pol-intra-asia_3
then permit
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding MGCP ALGs on page 261
 - MGCP ALG Configuration Overview on page 267

CHAPTER 15

RPC ALGs

- Understanding RPC ALGs on page 287
- Sun RPC ALGs on page 288
- Microsoft RPC ALGs on page 291

Understanding RPC ALGs

Junos OS supports basic Remote Procedure Call Application Layer Gateway (RPC ALG) services. RPC is a protocol that allows an application running in one address space to access the resources of applications running in another address space as if the resources were local to the first address space. The RPC ALG is responsible for RPC packet processing.

The RPC ALG in Junos OS supports the following services and features:

- Sun Microsystems RPC Open Network Computing (ONC)
- Microsoft RPC Distributed Computing Environment (DCE)
- Dynamic port negotiation
- Ability to allow and deny specific RPC services
- Static Network Address Translation (NAT) and source NAT (with no port translation)
- RPC applications in security policies

Use the RPC ALG if you need to run RPC-based applications such as NFS or Microsoft Outlook. The RPC ALG functionality is enabled by default.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - ALG Overview on page 169
 - Understanding Sun RPC ALGs on page 288
 - Understanding Microsoft RPC ALGs on page 291

Sun RPC ALGs

- Understanding Sun RPC ALGs on page 288
- Enabling Sun RPC ALGs (J-Web Procedure) on page 289
- Enabling Sun RPC ALGs (CLI Procedure) on page 289
- Sun RPC Services and Applications on page 289

Understanding Sun RPC ALGs

Sun Microsystems Remote Procedure Call (Sun RPC)—also known as Open Network Computing Remote Procedure Call (ONC RPC)—provides a way for a program running on one host to call procedures in a program running on another host. Because of the large number of RPC services and the need to broadcast, the transport address of an RPC service is dynamically negotiated based on the service's program number and version number. Several binding protocols are defined for mapping the RPC program number and version number to a transport address.

Junos OS supports the Sun RPC as a predefined service and allows and denies traffic based on a security policy you configure. The Application Layer Gateway (ALG) provides the functionality for Juniper Networks devices to handle the dynamic transport address negotiation mechanism of the Sun RPC and to ensure program number-based security policy enforcement. You can define a security policy to permit or deny all RPC requests, or to permit or deny by specific program number. The ALG also supports route mode and Network Address Translation (NAT) mode for incoming and outgoing requests.

When an application or a PC client calls a remote service, it needs to find the transport address of the service. In the case of TCP/UDP, the address is a port number. A typical procedure for this case is as follows:

1. The client sends the GETPORT message to the RPCBIND service on the remote machine. The GETPORT message contains the program number, and version and procedure number of the remote service it is attempting to call.
2. The RPCBIND service replies with a port number.
3. The client calls the remote service using the port number returned.
4. The remote service replies to the client.

A client also can use the CALLIT message to call the remote service directly, without determining the port number of the service. In this case, the procedure is as follows:

1. The client sends a CALLIT message to the RPCBIND service on the remote machine. The CALLIT message contains the program number and the version and procedure number of the remote service it attempting to call.
2. RPCBIND calls the service for the client.
3. RCPBIND replies to the client if the call has been successful. The reply contains the call result and the services's port number.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding RPC ALGs on page 287
 - Enabling Sun RPC ALGs (J-Web Procedure) on page 289
 - Enabling Sun RPC ALGs (CLI Procedure) on page 289
 - Understanding Sun RPC Services on page 290
 - Understanding Microsoft RPC ALGs on page 291

Enabling Sun RPC ALGs (J-Web Procedure)

The Sun RPC ALG is enabled by default and requires no configuration.

To disable or re-enable the RPC ALG:

1. Select **Configure>Security>ALG**.
2. Select the **Enable SUNRPC** check box.
3. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Sun RPC ALGs on page 288
 - Enabling Sun RPC ALGs (CLI Procedure) on page 289

Enabling Sun RPC ALGs (CLI Procedure)

The Sun RPC ALG is enabled by default and requires no configuration.

To disable the Sun RPC ALG, enter the following command:

```
user@host# set security alg sunrpc disable
```

To re-enable the Sun RPC ALG, enter the following command:

```
user@host# delete security alg sunrpc
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Sun RPC ALGs on page 288
 - Enabling Sun RPC ALGs (J-Web Procedure) on page 289

Sun RPC Services and Applications

- Understanding Sun RPC Services on page 290
- Customizing Sun RPC Applications (CLI Procedure) on page 290

Understanding Sun RPC Services

Predefined Sun RPC services include:

- **junos-sun-rpc-portmap-tcp**
- **junos-sun-rpc-portmap**
- **junos-sun-rpc-portmap-udp**

The Sun RPC ALG can be applied by using the following methods:

- ALG default application—Use one of the following predefined application sets for control and data connections in your policy:
 - **application-set junos-sun-rpc** (for control sessions)
 - **application-set junos-sun-rpc-portmap** (for data sessions)
- Default control application—Use the predefined control via **junos-sun-rpc**:
 - Create an application for data (**USER_DEFINED_DATA**). You can make a set of your own data (for example, **my_rpc_application_set**) and use it in the policy.
 - Use the predefined application set for control and customized data application in the policy:
 - **junos-sun-rpc**
 - **USER_DEFINED_DATA**
- Custom control and custom data application—Use a customized application:
 - Create an application for control (**USER_DEFINED_CONTROL**) and data (**USER_DEFINED_DATA**).
 - In the policy, use the user-defined application set for a control and customized data application:
 - **USER_DEFINED_CONTROL**
 - **USER_DEFINED_DATA**

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Sun RPC ALGs on page 288
 - Customizing Sun RPC Applications (CLI Procedure) on page 290
 - Understanding Microsoft RPC Services on page 293

Customizing Sun RPC Applications (CLI Procedure)

All Sun RPC applications can be customized by using a predefined application set.

For example, an application can be customized to open the control session only and not allow any data sessions:

```
application-set junos-sun-rpc {
  application junos-sun-rpc-tcp;
  application junos-sun-rpc-udp;
}
```

In the following example, the predefined application set allows data sessions only. It will not work without the control session:

```
application-set junos-sun-rpc-portmap {
  application junos-sun-rpc-portmap-tcp;
  application junos-sun-rpc-portmap-udp;
}
```

To customize all Sun RPC applications with predefined application sets, use both application sets in the policy:

```
application-set [junos-sun-rpc junos-sun-rpc-portmap]
```



NOTE: MS RPC applications are customized in the same way as SUN RPC applications.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Sun RPC ALGs on page 288
- Customizing Microsoft RPC Applications (CLI Procedure) on page 293

Microsoft RPC ALGs

- Understanding Microsoft RPC ALGs on page 291
- Enabling Microsoft RPC ALGs (J-Web Procedure) on page 292
- Enabling Microsoft RPC ALGs (CLI Procedure) on page 292
- Microsoft RPC Services and Applications on page 293
- Verifying the Microsoft RPC ALG Tables on page 293

Understanding Microsoft RPC ALGs

Microsoft Remote Procedure Call (MS RPC) is the Microsoft implementation of the Distributed Computing Environment (DCE) RPC. Like the Sun RPC, MS RPC provides a way for a program running on one host to call procedures in a program running on another host. Because of the large number of RPC services and the need to broadcast, the transport address of an RPC service is dynamically negotiated based on the service program's Universal Unique Identifier (UUID). The specific UUID is mapped to a transport address.

Junos OS devices running Junos OS support MS RPC as a predefined service and allow and deny traffic based on a policy you configure. The Application Layer Gateway (ALG) provides the functionality for Juniper Networks devices to handle the dynamic transport address negotiation mechanism of the MS RPC, and to ensure UUID-based security policy

enforcement. You can define a security policy to permit or deny all RPC requests, or to permit or deny by specific UUID number. The ALG also supports route mode and Network Address Translation (NAT) mode for incoming and outgoing requests.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding RPC ALGs on page 287
 - Enabling Microsoft RPC ALGs (J-Web Procedure) on page 292
 - Enabling Microsoft RPC ALGs (CLI Procedure) on page 292
 - Understanding Microsoft RPC Services on page 293
 - Understanding Sun RPC ALGs on page 288
 - Verifying the Microsoft RPC ALG Tables on page 293

Enabling Microsoft RPC ALGs (J-Web Procedure)

The MS RPC ALG is enabled by default and requires no configuration.

To disable or re-enable the Microsoft ALG:

1. Select **Configure>Security>ALG**.
2. Select the **Enable MSRPC** check box.
3. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Microsoft RPC ALGs on page 291
 - Enabling Microsoft RPC ALGs (CLI Procedure) on page 292
 - Verifying the Microsoft RPC ALG Tables on page 293

Enabling Microsoft RPC ALGs (CLI Procedure)

The MS RPC ALG is enabled by default and requires no configuration.

To disable the Microsoft RPC ALG, enter the following command:

```
user@host# set security alg msrpc disable
```

To re-enable the Microsoft RPC ALG, enter the following command:

```
user@host# delete security alg msrpc
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Microsoft RPC ALGs on page 291
 - Enabling Microsoft RPC ALGs (J-Web Procedure) on page 292
 - Verifying the Microsoft RPC ALG Tables on page 293

Microsoft RPC Services and Applications

- Understanding Microsoft RPC Services on page 293
- Customizing Microsoft RPC Applications (CLI Procedure) on page 293

Understanding Microsoft RPC Services

Predefined MS RPC services include:

- `junos-ms-rpc-portmap`
- `junos-ms-rpc-portmap-tcp`
- `junos-ms-rpc-portmap-udp`

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Microsoft RPC ALGs on page 291
 - Customizing Microsoft RPC Applications (CLI Procedure) on page 293
 - Understanding Sun RPC Services on page 290

Customizing Microsoft RPC Applications (CLI Procedure)

MS RPC applications are customized in the same way as SUN RPC applications.

MS RPC services in security policies are:

- `0e4a0156-dd5d-11d2-8c2f-00c04fb6bcde`
- `1453c42c-0fa6-11d2-a910-00c04f990f3b`
- `10f24e8e-0fa6-11d2-a910-00c04f990f3b`
- `1544f5e0-613c-11d1-93df-00c04fd7bd09`

The corresponding TCP/UDP ports are dynamic. To permit them, you use the following statement for each number:

```
set applications application-name term term-name uuid hex-number
```

The ALG maps the program numbers into dynamically negotiated TCP/UDP ports based on these four UUIDs and permits or denies the service based on a policy you configure.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Microsoft RPC Services on page 293
 - Customizing Sun RPC Applications (CLI Procedure) on page 290
 - Verifying the Microsoft RPC ALG Tables on page 293

Verifying the Microsoft RPC ALG Tables

- Purpose** To verify the Microsoft RPC ALG, display the Microsoft Universal Unique Identifier to Object ID (UUID-to-OID) mapping table. The Microsoft RPC ALG monitors packets on TCP port 135.

Action From the CLI, enter the **show security alg msrpc object-id-map** command.

```
user@host> show security alg msrpc object-id-map
UUID                                OID
1be617c0-31a5-11cf-a7d8-00805f48a135 0x80000020
e3514235-4b06-11d1-ab04-00c04fc2dcd2 0x80000002
67df7c70-0f04-11ce-b13f-00aa003bac6c 0x80000014
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Enabling Microsoft RPC ALGs (J-Web Procedure) on page 292
 - Enabling Microsoft RPC ALGs (CLI Procedure) on page 292
 - Customizing Microsoft RPC Applications (CLI Procedure) on page 293

PART 5

User Authentication

- Firewall User Authentication on page 297
- Infranet Authentication on page 327

Firewall User Authentication

- Firewall User Authentication Overview on page 297
- Pass-Through Authentication on page 298
- Web Authentication on page 304
- External Authentication on page 312
- Client Groups for Firewall Authentication on page 321
- Firewall Authentication Banner Customization on page 323

Firewall User Authentication Overview

A firewall user is a network user who must provide a username and password for authentication when initiating a connection across the firewall. Junos OS enables administrators to restrict and permit firewall users to access protected resources (different zones) behind a firewall based on their source IP address and other credentials.



NOTE: Junos OS also supports the administrator and Point-to-Point Protocol (PPP) user types. For more information, see the *Junos OS Administration Guide for Security Devices*.

After you define firewall users, you can create a policy that requires the users to authenticate themselves through one of two authentication schemes:

- **Pass-Through Authentication**—A host or a user from one zone tries to access resources on another zone. You must use an FTP, a Telnet, or an HTTP client to access the IP address of the protected resource and to get authenticated by the firewall. The device uses FTP, Telnet, or HTTP to collect username and password information, and subsequent traffic from the user or host is allowed or denied based on the result of this authentication.
- **Web Authentication**—Users try to connect, using HTTP, to an IP address on the device that is enabled for Web authentication; in this scenario, you do not use HTTP to get to the IP address of the protected resource. You are prompted for the username and password that are verified by the device. Subsequent traffic from the user or host to the protected resource is allowed or denied based on the result of this authentication.

Related Topics • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Understanding Pass-Through Authentication on page 298
- Understanding Web Authentication on page 305
- Understanding External Authentication Servers on page 312
- Understanding Client Groups for Firewall Authentication on page 321
- Understanding Firewall Authentication Banner Customization on page 323

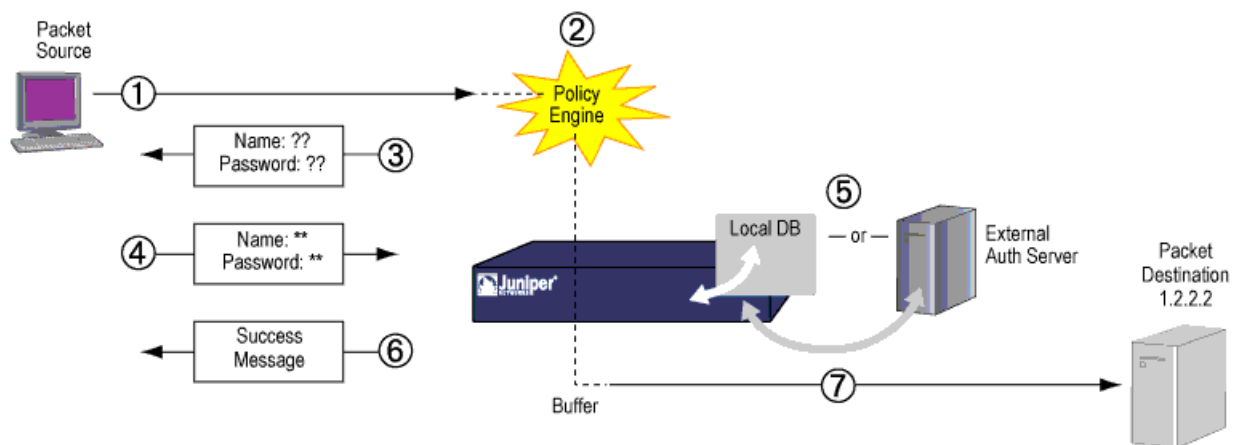
Pass-Through Authentication

- Understanding Pass-Through Authentication on page 298
- Example: Configuring Pass-Through Authentication on page 299

Understanding Pass-Through Authentication

With pass-through user authentication, when a user attempts to initiate an HTTP, an FTP, or a Telnet connection request that has a policy requiring authentication, the device intercepts the request and prompts the user to enter a username and password. Before granting permission, the device validates the username and password by checking them against those stored in the local database or on an external authentication server, as shown in Figure 32 on page 298.

Figure 32: Policy Lookup for a User



The steps in Figure 32 on page 298 are as follows:

1. A client user sends an FTP, an HTTP, or a Telnet packet to 1.2.2.2.
2. The device intercepts the packet, notes that its policy requires authentication from either the local database or an external authentication server, and buffers the packet.
3. The device prompts the user for login information through FTP, HTTP, or Telnet.
4. The user replies with a username and password.
5. The device either checks for an authentication user account on its local database or it sends the login information to the external authentication server as specified in the policy.

6. Finding a valid match (or receiving notice of such a match from the external authentication server), the device informs the user that the login has been successful.
7. The device forwards the packet from its buffer to its destination IP address 1.2.2.2.

After the device authenticates a user at a particular source IP address, it subsequently permits traffic—as specified in the policy requiring authentication through pass through—from any other user at that same address. This might be the case if the user originates traffic from behind a NAT device that changes all original source addresses to a single translated address.



NOTE: The pass-through user authentication method is recommended in situations when security has a higher priority than convenience. This authentication method applies only to the session and child sessions matching the policy that triggered it. You can apply this method on Internet-facing links, if used with caution.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Firewall User Authentication Overview on page 297
- Understanding Web Authentication on page 305
- Example: Configuring Pass-Through Authentication on page 299

Example: Configuring Pass-Through Authentication

This example shows how to configure pass-through authentication for a firewall.

- Requirements on page 299
- Overview on page 299
- Configuration on page 300
- Verification on page 303

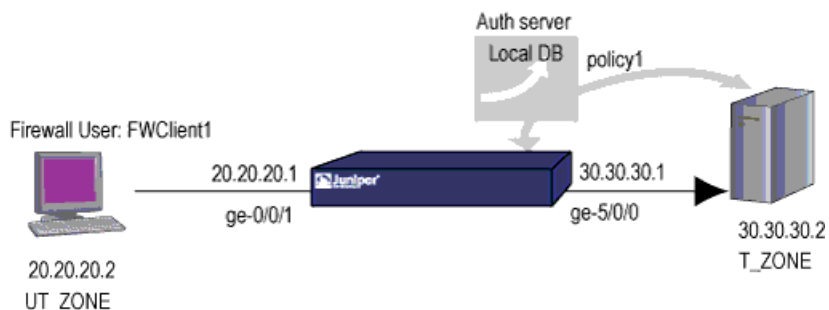
Requirements

Before you begin, define firewall users. See “Firewall User Authentication Overview” on page 297.

Overview

Pass-through firewall user authentication occurs when the client is trying to access a destination on another zone using FTP, Telnet, or HTTP. After authenticating successfully, the firewall acts as a proxy for an FTP, a Telnet, or an HTTP server so that it can first authenticate the user before allowing access to the actual FTP, Telnet, or HTTP server behind the firewall. Figure 33 on page 300 shows the topology used in this example.

Figure 33: Configuring Pass-Through Firewall Authentication



Configuration

CLI Quick Configuration

To quickly configure pass-through authentication, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.1/24
set interfaces ge-5/0/0 unit 0 family inet address 30.30.30.1/24
set access profile FWAUTH client FWClient1 firewall-user password pwd
set access firewall-authentication pass-through default-profile FWAUTH
set access firewall-authentication pass-through telnet banner success "WELCOME
TO JUNIPER TELNET SESSION"
set security zones security-zone UT-ZONE host-inbound-traffic system-services all
set security zones security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic
protocols all
set security zones security-zone T-ZONE host-inbound-traffic system-services all
set security zones security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic
protocols all
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match
source-address any
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match
destination-address any
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match application
junos-telnet
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 then permit
firewall-authentication pass-through client-match FWClient1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure pass-through authentication:

1. Configure two interfaces and assign IP addresses to them.



NOTE: For this example, it is optional to assign two addresses to the interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.1/24
user@host# set interfaces ge-5/0/0 unit 0 family inet address 30.30.30.1/24
```

2. Create the FWAUTH access profile for the FWClient1 user, specify the user's password, and define a success banner for Telnet sessions.

```
[edit access]
user@host# set access profile FWAUTH client FWClient1 firewall-user password
pwd
user@host# set firewall-authentication pass-through default-profile FWAUTH
user@host# set firewall-authentication pass-through telnet banner success
"WELCOME TO JUNIPER TELNET SESSION"
```

3. Configure security zones.



NOTE: For this example, it is optional to configure a second interface for a security zone.

```
[edit security zones]
user@host# set security-zone UT-ZONE host-inbound-traffic system-services all
user@host# set security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic
protocols all
user@host# set security-zone T-ZONE host-inbound-traffic system-services all
user@host# set security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic
protocols all
```

4. Assign security policy P1 to the security zones.

```
[edit security policies]
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match
source-address any
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match
destination-address any
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match application
junos-telnet
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 then permit
firewall-authentication pass-through client-match FWClient1
```

5. Use Telnet to authenticate the FWClient1 firewall user to host2.

```
regress@FWClient1# run telnet 30.30.30.2
Trying 30.30.30.2...
Connected to 30.30.30.2.
Escape character is '^]'.
Firewall User Authentication
Username: FWClient1
Password:***
WELCOME TO JUNIPER TELNET SESSION
Host1 (tty0)
login: regress
Password:
--- JUNOS 10.1R1.1 built 2009-10-12 13:30:18 UTC
%
```

Results From configuration mode, confirm your configuration by entering these commands:

- show interfaces
- show access

- **show security zones**
- **show security policies**

If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```

user@host# show interfaces
...
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 20.20.20.1/24;
        }
    }
}
ge-5/0/0 {
    unit 0 {
        family inet {
            address 30.30.30.1/24;
        }
    }
}
...

user@host# show access
profile FWAUTH {
    authentication-order password;
    client FWClient1 {
        firewall-user {
            password "$9$XHhXVYGDkf5F"; ## SECRET-DATA
        }
    }
}
firewall-authentication {
    pass-through {
        default-profile FWAUTH;
        telnet {
            banner {
                success "WELCOME TO JUNIPER TELNET SESSION";
            }
        }
    }
}

user@host# show security zones
...
}
security-zone UT-ZONE {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0 {
            host-inbound-traffic {

```

```

        protocols {
            all;
        }
    }
}
}
security-zone T-ZONE {
    host-inbound-traffic {
        system-services {
            all;
        }
    }
    interfaces {
        ge-5/0/0.0 {
            host-inbound-traffic {
                protocols {
                    all;
                }
            }
        }
    }
}

user@host# show security policies
...
from-zone UT-ZONE to-zone T-ZONE {
    policy P1 {
        match {
            source-address any;
            destination-address any;
            application junos-telnet;
        }
        then {
            permit {
                firewall-authentication {
                    pass-through {
                        client-match FWClient1;
                    }
                }
            }
        }
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table on page 303

Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table

Purpose Display firewall authentication user history and verify the number of firewall users who successfully authenticated and firewall users who failed to log in.

Action From operational mode, enter these **show** commands:

```
user@host> show security firewall-authentication history
user@host> show security firewall-authentication history identifier 1
user@host> show security firewall-authentication users
user@host> show security firewall-authentication users identifier 3
```

For more information, see the *Junos OS CLI Reference*.

```
user@host> show security firewall-authentication history
History of firewall authentication data:
Authentications: 2
Id Source Ip Date Time Duration Status User
1 20.20.20.2 2010-10-12 21:24:02 0:00:24 Failed FWClient1
2 20.20.20.2 2010-10-12 21:24:48 0:00:22 Success FWClient1
```

```
user@host> show security firewall-authentication history identifier 1
Username: FWClient1
Source IP: 20.20.20.2
Authentication state: Success
Authentication method: Pass-through using Telnet
Access start date: 2010-10-12
Access start time: 21:24:02
Duration of user access: 0:00:24
Source zone: UT-ZONE
Destination zone: T-ZONE
Access profile: FWAUTH
Bytes sent by this user: 0
Bytes received by this user: 2660
```

```
user@host> show security firewall-authentication users
Firewall authentication data:
Total users in table: 1
Id Source Ip Src zone Dst zone Profile Age Status User
4 20.20.20.2 UT-ZONE T-ZONE FWAUTH 1 Success FWClient1
```

```
user@host> show security firewall-authentication users identifier 3
Username: FWClient1
Source IP: 20.20.20.2
Authentication state: Success
Authentication method: Pass-through using Telnet
Age: 3
Access time remaining: 9
Source zone: UT-ZONE
Destination zone: T-ZONE
Access profile: FWAUTH
Interface Name: ge-0/0/1.0
Bytes sent by this user: 0
Bytes received by this user: 1521
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Firewall User Authentication Overview on page 297
 - Understanding Pass-Through Authentication on page 298

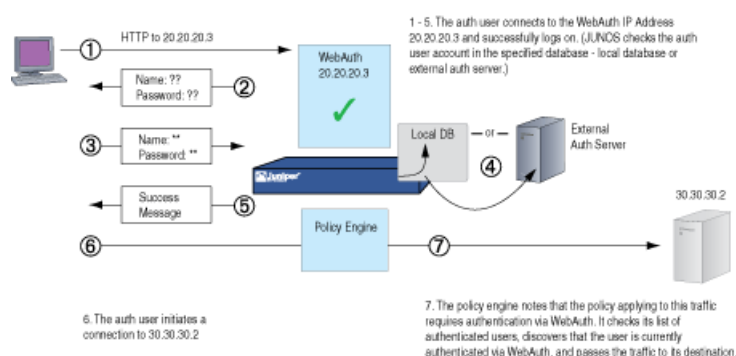
Web Authentication

- Understanding Web Authentication on page 305
- Example: Configuring Web Authentication on page 306

Understanding Web Authentication

Web authentication is an alternative to pass-through user authentication. Instead of pointing to the resource that you want to connect to from your client browser, you point the browser to an IP address on the device that is enabled for Web authentication. This initiates an HTTP session to the IP address hosting the Web authentication feature on the device. The device then prompts you for your username and password and caches the result in the device. Later, when traffic encounters a Web authentication policy, you are allowed or denied access based on the prior Web authentication results, as shown in Figure 34 on page 305.

Figure 34: Web Authentication Example



Follow these Web authentication guidelines:

- You can leave the default Web authentication server as the local database or you can choose an external authentication server for the role. The default Web authentication profile determines if the user authenticates using the local database or the external authentication server. An access profile stores usernames and passwords of users or points to external authentication servers where such information is stored.
- The Web authentication address must be in the same subnet as the interface that you want to use to host it. For example, if you want authentication users to connect using Web authentication through ethernet3, which has IP address 1.1.1.1/24, then you can assign Web authentication an IP address in the 1.1.1.0/24 subnet.
- You can put a Web authentication address in the same subnet as the IP address of any physical interface or virtual security interface (VSI). (For information about different types of interfaces, see “Security Zones and Interfaces Overview” on page 85.)
- You can put Web authentication addresses on multiple interfaces.
- After a device authenticates a user at a particular source IP address, it subsequently permits traffic—as specified in the policy requiring authentication through Web authentication—from any other user at that same address. This might be the case if the user originates traffic from behind a NAT device that changes all original source addresses to a single translated address.
- With Web authentication enabled, any HTTP traffic to the IP address will get the Web authentication login page instead of the administrator login page. Disabling this option

will show the administrator login page (assuming that **[system services web-management HTTP]** is enabled).

- We recommend that you have a separate primary or preferred IP address, if an address is used for Web authentication.



NOTE: The Web authentication method is recommended in situations when the client devices are immediately adjacent to the security gateway and there is high assurance that the client devices are not multiuser hosts. This authentication method is best applied to wireless links and DMZ, or conference room links.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Firewall User Authentication Overview on page 297
- Understanding Pass-Through Authentication on page 298
- Example: Configuring Web Authentication on page 306

Example: Configuring Web Authentication

This example shows how to enable Web authentication and set up a policy that allows access to a user when traffic encounters a policy that has Web authentication enabled.

- Requirements on page 306
- Overview on page 306
- Configuration on page 307
- Verification on page 311

Requirements

Before you begin:

- Define firewall users. See “Firewall User Authentication Overview” on page 297.
- Add the Web authentication HTTP flag under the interface’s address hierarchy to enable Web authentication.

Overview

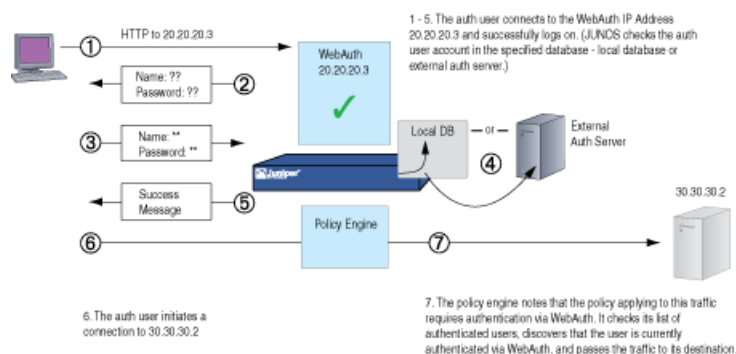
To enable Web authentication, you must specify the IP address of the device hosting the HTTP session. These settings are used if the firewall user accessing a protected resource wants to be authenticated by directly accessing the webserver or Web authentication. The following instructions show how to set up a policy that allows access to the FWClient1 user when traffic encounters a policy that has Web authentication enabled (Policy-W). (See Figure 35 on page 307.) In this example, FWClient1 has already authenticated through the Web authentication login page.

The FWClient1 firewall user does the following to get authenticated:

- a. Points the browser to the Web authentication IP (20.20.20.1) to get authenticated first

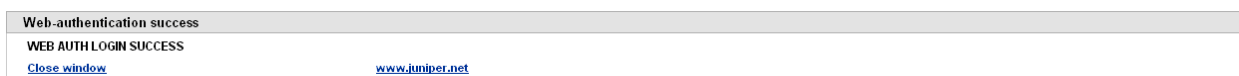
- b. Starts traffic to access resources specified by the policy-W policy

Figure 35: Web Authentication Example



When you configure the device as described in these instructions and the user successfully authenticates, the screen illustrated in Figure 36 on page 307 appears.

Figure 36: Web Authentication Success Banner



Configuration

CLI Quick Configuration To quickly configure Web authentication as illustrated in Figure 35 on page 307, copy the following commands and paste them into the CLI:

```
[edit]
set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.1/24
set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.3/24 web-authentication
http
set interfaces fe-5/0/0 unit 0 family inet address 30.30.30.1/24
set access profile WEBAUTH client FWClient1 firewall-user password pwd
set access firewall-authentication web-authentication default-profile WEBAUTH
set access firewall-authentication web-authentication banner success "WEB AUTH
LOGIN SUCCESS"
set security zones security-zone UT-ZONE host-inbound-traffic system-services all
set security zones security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic
protocols all
set security zones security-zone T-ZONE host-inbound-traffic system-services all
set security zones security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic
protocols all
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match
source-address any
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match
destination-address any
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 match application
any
set security policies from-zone UT-ZONE to-zone T-ZONE policy P1 then permit
firewall-authentication web-authentication client-match FWClient1
set system services web-management http interface ge-0/0/1.0
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure Web authentication:

1. Configure two interfaces and assign IP addresses to them.



NOTE: For this example, it is optional to assign two addresses to the interfaces.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.1/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.3/24
web-authentication http
user@host# set interfaces fe-5/0/0 unit 0 family inet address 30.30.30.1/24
```

2. Create the WEBAUTH access profile for the FWClient1 user, specify the user's password, and define a success banner.

```
[edit access]
user@host# set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.1/24
user@host# set firewall-authentication web-authentication default-profile
WEBAUTH
user@host# set firewall-authentication web-authentication banner success "WEB
AUTH LOGIN SUCCESS"
```

3. Configure security zones.



NOTE: For this example, it is optional to configure a second interface for a security zone.

```
[edit security zones]
user@host# set security-zone UT-ZONE host-inbound-traffic system-services all
user@host# set security-zone UT-ZONE interfaces ge-0/0/1.0 host-inbound-traffic
protocols all
user@host# set security-zone T-ZONE host-inbound-traffic system-services all
user@host# set security-zone T-ZONE interfaces ge-5/0/0.0 host-inbound-traffic
protocols all
```

4. Assign security policy P1 to the security zones.

```
[edit security policies]
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match
source-address any
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match
destination-address any
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 match application
any
user@host# set from-zone UT-ZONE to-zone T-ZONE policy P1 then permit
firewall-authentication web-authentication client-match FWClient1
```

5. Activate the HTTP daemon on your device.

```
[edit]
user@host# set system services web-management http interface ge-0/0/1.0
```

Results From configuration mode, confirm your configuration by entering these commands:

- **show interfaces**
- **show access**
- **show security zones**
- **show security policies**
- **show system services**

If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
user@host# show interfaces
...
}
ge-0/0/1{
    unit 0 {
        family inet {
            address 20.20.20.1/24 {
            address 20.20.20.3/24 {
                web-authentication http;
            }
        }
    }
}
fe-5/0/0 {
    unit 0 {
        family inet {
            address 30.30.30.1/24;
        }
    }
}
...

user@host# show access
profile WEBAUTH {
    client FWClient1 {
        firewall-user {
            password "$9$XHhXVYGdkf5F"; ## SECRET-DATA
        }
    }
}
firewall-authentication {
    web-authentication {
        default-profile WEBAUTH;
        banner {
            success "WEB AUTH LOGIN SUCCESS";
        }
    }
}
```

```
}

user@host# show security zones
...
}
security-zone UT-ZONE {
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  interfaces {
    ge-0/0/1.0 {
      host-inbound-traffic {
        protocols {
          all;
        }
      }
    }
  }
}
security-zone T-ZONE {
  host-inbound-traffic {
    system-services {
      all;
    }
  }
  interfaces {
    ge-5/0/0.0 {
      host-inbound-traffic {
        protocols {
          all;
        }
      }
    }
  }
}

user@host# show security policies
...
from-zone UT-ZONE to-zone T-ZONE {
  policy P1 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit {
        firewall-authentication {
          web-authentication {
            client-match FWClient1;
          }
        }
      }
    }
  }
}

user@host# show system services
...
ftp;
```

```
ssh;
telnet;
web-management {
    http {
        interface g-0/0/1.0;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table on page 311

Verifying Firewall User Authentication and Monitoring Users and IP Addresses in the Authentication Table

Purpose Display firewall authentication user history and verify the number of firewall users who successfully authenticated and firewall users who failed to log in.

Action From operational mode, enter these **show** commands:

```
user@host> show security firewall-authentication history
user@host> show security firewall-authentication history identifier 1
user@host> show security firewall-authentication users
user@host> show security firewall-authentication users identifier 3
```

```
user@host> show security firewall-authentication history
History of firewall authentication data:
Authentications: 1
Id Source Ip Date Time Duration Status User
5 20.20.20.2      2010-04-24 01:08:57 0:10:30    Success  FWClient1
```

```
user@host> show security firewall-authentication history identifier 1
Username: FWClient1
Source IP: 20.20.20.2
Authentication state: Success
Authentication method: Web-authentication
Access start date: 2010-10-12
Access start time: 21:24:02
Duration of user access: 0:00:24
Source zone: N/A
Destination zone: N/A
Access profile: WEBAUTH
Bytes sent by this user: 0
Bytes received by this user: 2660
```

```
user@host> show security firewall-authentication users
Firewall authentication data:
Total users in table: 1
Id Source Ip Src zone Dst zone Profile Age Status User
4 20.20.20.2      N/A  N/A  WEBAUTH      1 Success  FWClient1
```

```
user@host> show security firewall-authentication users identifier 3
Username: FWClient1
Source IP: 20.20.20.2
Authentication state: Success
Authentication method: Web-authentication
Age: 3
```

Access time remaining: 9
Source zone: N/A
Destination zone: N/A
Access profile: WEBAUTH
Interface Name: ge-0/0/1.0
Bytes sent by this user: 0
Bytes received by this user: 1521

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Web Authentication on page 305
 - Understanding Firewall Authentication Banner Customization on page 323
 - Security Zones and Interfaces Overview on page 85

External Authentication

- Understanding External Authentication Servers on page 312
- Example: Configuring RADIUS and LDAP User Authentication on page 314
- Example: Configuring SecurID User Authentication on page 317
- Example: Deleting the SecurID Node Secret File on page 320

Understanding External Authentication Servers

AAA provides an extra level of protection and control for user access in the following ways:

- Authentication determines the firewall user.
- Authorization determines what the firewall user can do.
- Accounting determines what the firewall user did on the network.

You can use authentication alone or with authorization and accounting. Authorization always requires a user to be authenticated first. You can use accounting alone, or with authentication and authorization.

Once the user's credentials are collected, they are processed using firewall user authentication, which supports the following types of servers:

- Local authentication and authorization
- RADIUS authentication and authorization (compatible with Juniper SteelBelted Radius server)
- LDAP authentication only (supports LDAP version 3 and compatible with Windows AD)
- SecurID authentication only (using an RSA SecurID external authentication server)



NOTE: Junos OS also supports administrative authentication using local, RADIUS, and TACACS+ servers. For more information on administrative authentication, see the *Junos OS Administration Guide for Security Devices*.

This topic includes the following sections:

- Understanding SecurID User Authentication on page 313

Understanding SecurID User Authentication

SecurID is an authentication method that allows users to enter either static or dynamic passwords as their credentials. A dynamic password is a combination of a user's PIN and a randomly generated token that is valid for a short period of time, approximately one minute. A static password is set for the user on the SecurID server. For example, the SecurID server administrator might set a temporary static password for a user who lost his or her SecurID token.

When a user attempts to access a resource protected by a policy and SecurID is configured in the profile **authentication-order** parameter as either the only authentication mode or the first one to be used, the device forwards the user's credentials to the SecurID server for authentication. If the user enters valid values, the user is allowed access to the requested resource.



NOTE: The SecurID server includes a feature that presents a user with a challenge if the user provides wrong credentials repeatedly. However, Junos OS does not support the challenge feature. Instead, the SecurID server administrator must resynchronize the RSA token for the user.

For SecurID, you configure information about the Juniper Networks device on the SecurID server and this information is exported to a file called **sdconf.rec**.

To install the **sdconf.rec** file on the device, you must use an out-of-band method such as FTP. Install the file in a directory whose files are not deleted regularly. Do not put it in a temporary directory. For example, you might install it in **/var/db/secureid/server1/sdconf.rec**.

The **sdconf.rec** file contains information that provides the Juniper Networks device with the address of the SecurID server. You do not need to configure this information explicitly when you configure the SecurID server to be used as the external authentication server.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Firewall User Authentication Overview on page 297
- Example: Configuring RADIUS and LDAP User Authentication on page 314
- Example: Configuring SecurID User Authentication on page 317
- Example: Deleting the SecurID Node Secret File on page 320

Example: Configuring RADIUS and LDAP User Authentication

This example shows how to configure a device for external authentication.

- Requirements on page 314
- Overview on page 314
- Configuration on page 314
- Verification on page 317

Requirements

Before you begin, create an authentication user group.

Overview

You can put several user accounts together to form a user group, which you can store on the local database or on a RADIUS, an LDAP, or a SecurID server. When you reference an authentication user group and an external authentication server in a policy, the traffic matching the policy provokes an authentication check.

This example shows how access profile Profile-1 is configured for external authentication. Two RADIUS servers and one LDAP server are configured in the access profile. However, the order of authentication specifies RADIUS server only, so if the RADIUS server authentication fails, then the firewall user fails to authenticate. The local database is not accessed.



NOTE: If the firewall clients are authenticated by the RADIUS server, then the group-membership VSA returned by the RADIUS server should contain alpha, beta, or gamma client groups in the RADIUS server configuration or in the access profile, Profile-1. Access profiles store usernames and passwords of users or point to external authentication servers where such information is stored.

Configuration

CLI Quick Configuration

To quickly configure a device for external authentication, copy the following commands and paste them into the CLI:

```
[edit]
set access profile Profile-1 authentication-order radius
set access profile Profile-1 client Client-1 client-group alpha
set access profile Profile-1 client Client-1 client-group beta
set access profile Profile-1 client Client-1 client-group gamma
set access profile Profile-1 client Client-1 firewall-user password pwd
set access profile Profile-1 client Client-2 client-group alpha
set access profile Profile-1 client Client-2 client-group beta
set access profile Profile-1 client Client-2 firewall-user password pwd
set access profile Profile-1 client Client-3 firewall-user password pwd
set access profile Profile-1 client Client-4 firewall-user password pwd
set access profile Profile-1 session-options client-group alpha
set access profile Profile-1 session-options client-group beta
set access profile Profile-1 session-options client-group gamma
set access profile Profile-1 session-options client-idle-timeout 255
set access profile Profile-1 session-options client-session-timeout 4
```

```

set access profile Profile-1 ldap-options base-distinguished-name
CN=users,DC=junos,DC=juniper,DC=net
set access profile Profile-1 ldap-options search search-filter sAMAccountName=
set access profile Profile-1 ldap-options search admin-search distinguished-name
cn=administrator,cn=users,dc=junos,dc=juniper,dc=net
set access profile Profile-1 ldap-options search admin-search password juniper
set access profile Profile-1 ldap-server 3.3.3.3
set access profile Profile-1 radius-server 4.4.4.4 secret juniper
set access profile Profile-1 radius-server 4.4.4.4 retry 10
set access profile Profile-1 radius-server 5.5.5.5 secret juniper

```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a device for external authentication:

1. Specify the RADIUS server for external authentication order. This restricts firewall users to authenticate through the RADIUS server only. If the RADIUS server authentication fails and the default password (local database) option is not specified, the firewall user is locked out.

```

[edit]
user@host# set access profile Profile-1 authentication-order radius

```

2. Configure Client1-4 firewall users and assign the Client-1 firewall user to client groups alpha, beta, and gamma, and assign the Client-2 firewall user to client groups alpha and beta.

```

[edit access profile Profile-1]
user@host# set client Client-1 client-group alpha
user@host# set client Client-1 client-group beta
user@host# set client Client-1 client-group gamma
user@host# set client Client-1 firewall-user password pwd
user@host# set client Client-2 client-group alpha
user@host# set client Client-2 client-group beta
user@host# set client Client-2 firewall-user password pwd
user@host# set client Client-3 firewall-user password pwd
user@host# set client Client-4 firewall-user password pwd

```

3. Configure client groups in the session options.

```

[edit access profile Profile-1]
user@host# set session-options client-group alpha
user@host# set session-options client-group beta
user@host# set session-options client-group gamma
user@host# set session-options client-idle-timeout 255
user@host# set session-options client-session-timeout 4

```

4. Configure the IP address for the LDAP server and server options.

```

[edit access profile Profile-1]
user@host# set ldap-options base-distinguished-name
CN=users,DC=junos,DC=juniper,DC=net
user@host# set ldap-options search search-filter sAMAccountName=
user@host# set ldap-options search admin-search password juniper

```

```

user@host# set ldap-options search admin-search distinguished-name
cn=administrator,cn=users,dc=junos,dc=juniper,dc=net
user@host# set ldap-server 3.3.3.3

```

5. Configure the IP addresses for the two RADIUS servers.

```

[edit access profile Profile-1]
user@host# set radius-server 4.4.4.4 secret juniper
user@host# set radius-server 4.4.4.4 retry 10
user@host# set radius-server 5.5.5.5 secret juniper

```

Results From configuration mode, confirm your configuration by entering the **show access profile Profile-1** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

user@host# show access profile Profile-1
authentication-order radius;
client Client-1 {
  client-group [ alpha beta gamma ];
  firewall-user {
    password "$9$jpmT9A0REyn6yl"; ## SECRET-DATA
  }
}
client Client-2 {
  client-group [ alpha beta ];
  firewall-user {
    password "$9$IMVRyK7-w4oG-d"; ## SECRET-DATA
  }
}
client Client-3 {
  firewall-user {
    password "$9$GfUkPn/tB1h9C"; ## SECRET-DATA
  }
}
client Client-4 {
  firewall-user {
    password "$9$JuZi.FnCOOR/9"; ## SECRET-DATA
  }
}
session-options {
  client-group [ alpha beta gamma ];
  client-idle-timeout 255;
  client-session-timeout 4;
}
ldap-options {
  base-distinguished-name CN=users,DC=junos,DC=juniper,DC=net;
  search {
    search-filter sAMAccountName=;
    admin-search {
      distinguished-name
cn=administrator,cn=users,dc=junos,dc=juniper,dc=net;
      password "$9$PFF/01hleWB1X7"; ## SECRET-DATA
    }
  }
}
ldap-server {
  3.3.3.3;
}
radius-server {

```

```

4.4.4.4 {
    secret "$9$Q5WMF3/At0IRc"; ## SECRET-DATA
    retry 10;
}
5.5.5.5 {
    secret "$9$YUg4JUDHmPT"; ## SECRET-DATA
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 317

Troubleshooting with Logs

Purpose Use these logs to identify any issues.

Action From operational mode, enter the **show log messages** command and the **show log dcd** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding External Authentication Servers on page 312

Example: Configuring SecurID User Authentication

This example shows how to configure SecurID as the external authentication server.

- Requirements on page 317
- Overview on page 317
- Configuration on page 318
- Verification on page 319

Requirements

Before you begin:

- Create an authentication user group.
- Specify that Server-1 is to be used as the SecurID server and that its configuration file resides on the device in the `/var/db/secuid/Server-1/sdconf.rec` file. From configuration mode, enter this command:

```

user@host# set access securid-server Server-1 configuration-file
"/var/db/secuid/Server-1/sdconf.rec"

```

Overview

SecurID is an authentication method that allows users to enter either static or dynamic passwords as their credentials. A dynamic password is a combination of a user's PIN and a randomly generated token that is valid for a short period of time, approximately one minute. A static password is set for the user on the SecurID server. For example, the

SecurID server administrator might set a temporary static password for a user who lost his or her SecurID token.

When a user attempts to access a resource protected by a policy and SecurID is configured in the profile **authentication-order** parameter as either the only authentication mode or the first one to be used, the device forwards the user's credentials to the SecurID server for authentication. If the user enters valid values, the user is allowed access to the requested resource.

Configuration

CLI Quick Configuration

To quickly configure SecurID as the external authentication server, copy the following commands and paste them into the CLI:

```
[edit]
set access profile Profile-2 authentication-order securid
set access profile Profile-2 client Client-1 client-group alpha
set access profile Profile-2 client Client-1 client-group beta
set access profile Profile-2 client Client-1 client-group gamma
set access profile Profile-2 client Client-1 firewall-user password pwd
set access profile Profile-2 client Client-2 client-group alpha
set access profile Profile-2 client Client-2 client-group beta
set access profile Profile-2 client Client-2 firewall-user password pwd
set access profile Profile-2 client Client-3 firewall-user password pwd
set access profile Profile-2 client Client-4 firewall-user password pwd
set access profile Profile-2 session-options client-group alpha
set access profile Profile-2 session-options client-group beta
set access profile Profile-2 session-options client-group gamma
set access profile Profile-2 session-options client-idle-timeout 255
set access profile Profile-2 session-options client-session-timeout 4
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure SecurID as the external authentication server:

1. For the Profile-2 profile, configure SecurID as the server to be used for external authentication. This restricts firewall users to authenticate through the SecurID server only. If the SecurID server authentication fails, the firewall user is locked out:

```
[edit]
user@host# set access profile Profile-2 authentication-order securid
```

To share a single SecurID server across multiple profiles, for each profile set the **authentication-order** parameter to include **securid** as the authentication mode.

2. Configure Client1-4 firewall users and assign the Client-1 firewall user to client groups alpha, beta, and gamma, and assign the Client-2 firewall user to client groups alpha and beta.

```
[edit access profile Profile-2]
user@host# set client Client-1 client-group alpha
user@host# set client Client-1 client-group beta
user@host# set client Client-1 client-group gamma
user@host# set client Client-1 firewall-user password pwd
user@host# set client Client-2 client-group alpha
```

```

user@host# set client Client-2 client-group beta
user@host# set client Client-2 firewall-user password pwd
user@host# set client Client-3 firewall-user password pwd
user@host# set client Client-4 firewall-user password pwd

```

3. Configure client groups in the session options.

```

[edit access profile Profile-2]
user@host# set session-options client-group alpha
user@host# set session-options client-group beta
user@host# set session-options client-group gamma
user@host# set session-options client-idle-timeout 255
user@host# set session-options client-session-timeout 4

```

Results From configuration mode, confirm your configuration by entering the **show access profile Profile-2** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

user@host# show access profile Profile-2
authentication-order securid;
client Client-1 {
  client-group [ alpha beta gamma ];
  firewall-user {
    password "$9$jpmT9A0REyn6yl"; ## SECRET-DATA
  }
}
client Client-2 {
  client-group [ alpha beta ];
  firewall-user {
    password "$9$IMVRyK7-w4oG-d"; ## SECRET-DATA
  }
}
client Client-3 {
  firewall-user {
    password "$9$GfukPn/tB1h9C"; ## SECRET-DATA
  }
}
client Client-4 {
  firewall-user {
    password "$9$JuZi.FnC00R/9"; ## SECRET-DATA
  }
}
session-options {
  client-group [ alpha beta gamma ];
  client-idle-timeout 255;
  client-session-timeout 4;
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Troubleshooting with Logs on page 319

Troubleshooting with Logs

Purpose Use these logs to identify any issues.

Action From operational mode, enter the **show log messages** command and the **show log dcd** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding External Authentication Servers on page 312
 - Example: Deleting the SecurID Node Secret File on page 320

Example: Deleting the SecurID Node Secret File

This example shows how to delete the node secret file.

- Requirements on page 320
- Overview on page 320
- Configuration on page 320
- Verification on page 321

Requirements

Before you begin, confirm that it is necessary to delete the SecurID node secret file.

Overview

When the Juniper Networks device initially communicates successfully with the SecurID server, a node secret file is created for it automatically. The file is created as a result of the handshake between the Juniper Networks device and the SecurID server after the software authenticates the first user successfully. All subsequent communication between the Juniper Networks device and the SecurID server relies on this secret as a representation of trust between the two nodes instead of repeating the handshake with each authentication request.

Under normal circumstances you should not delete the node secret file. In the rare case that you must do so, for example, to debug a serious problem, you can use the **clear** command to remove the file.



WARNING: If you delete the file, you must deselect a box on the SecurID server to indicate that the node secret file for the Juniper Networks device and the SecurID server no longer exists. Otherwise, authentication attempts will fail.

Configuration

Step-by-Step Procedure

To delete the node secret file:

1. Use the **clear** command to remove the node secret file. During subsequent user authentication, the device reestablishes a shared secret with the SecurID server and re-creates the node secret file. From operational mode, enter the **clear network-access** command to clear the **securid-node-secret-file** for the Juniper Networks device.

```
user@host> clear network-access securid-node-secret-file
```


2. From operational mode, confirm your deletion by entering the **show network-access securid-node-secret-file** command. If the output does not display, repeat the instructions in this example to correct it.

```
user@host> show network-access securid-node-secret-file
```

Verification

Verify the deletion by entering the **show network-access securid-node-secret-file** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding External Authentication Servers on page 312
 - Example: Configuring SecurID User Authentication on page 317

Client Groups for Firewall Authentication

- Understanding Client Groups for Firewall Authentication on page 321
- Example: Configuring Local Users for Client Groups on page 322

Understanding Client Groups for Firewall Authentication

To manage a number of firewall users, you can create user or client groups and store the information either on the local Juniper Networks device or on an external RADIUS or LDAP server.

A client group is a list of groups to which the client belongs. As with client-idle timeout, a client group is used only if the external authentication server does not return a value in its response (for example, LDAP servers do not return such information).

The RADIUS server sends the client's group information to the Juniper Networks device using Juniper VSA (46). The client-match portion of the policy accepts a string that can either be the username or groupname the client belongs to.

The reason to have a single database for different types of clients (except admins) is based on the assumption that a single client can be of multiple types. For example, a firewall user client can also be an L2TP client.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Firewall User Authentication Overview on page 297
 - Example: Configuring Local Users for Client Groups on page 322
 - Example: Configuring a Default Client Group for All Users

Example: Configuring Local Users for Client Groups

This example shows how to configure a local user for client groups in a profile.

- Requirements on page 322
- Overview on page 322
- Configuration on page 322
- Verification on page 323

Requirements

Before you begin, create an access profile called Managers.

Overview

A client group is a list of groups to which the client belongs. As with client-idle timeout, a client group is used only if the external authentication server does not return a value in its response (for example, LDAP servers do not return such information).

This example shows how to configure a local user called Client-1 for client groups G1, G2, and G3 in a profile called Managers. Within this example, client groups are configured for a client. If a client group is not defined for the client, then the client group under the **access profile session-options** hierarchy is used.

Configuration

CLI Quick Configuration

To quickly configure a local user for client groups in a profile, copy the following commands and paste them into the CLI:

```
[edit]
set access profile Managers client Client-1 client-group G1
set access profile Managers client Client-1 client-group G2
set access profile Managers client Client-1 client-group G3
set access profile Managers client Client-1 firewall-user password pwd
set access profile Managers session-options client-group G1
set access profile Managers session-options client-group G2
set access profile Managers session-options client-group G3
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a local user for client groups in a profile:

1. Configure the firewall user and assign client groups to it.

```
[edit access profile Managers]
user@host# set client Client-1 client-group G1
user@host# set client Client-1 client-group G2
user@host# set client Client-1 client-group G3
user@host# set client Client-1 firewall-user password pwd
```

2. Configure client groups in the session options.

```
[edit access profile Managers]
user@host# set session-options client-group G1
```

```
user@host# set session-options client-group G2
user@host# set session-options client-group G3
```

Results Confirm your configuration by entering the **show access profile Managers** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

[edit]

```
user@host# show access profile Managers
```

```
client Client-1 {
  client-group [ G1 G2 G3 ];
  firewall-user {
    password "$9$jpmT9A0REyn6yl"; ## SECRET-DATA
  }
}
session-options {
  client-group [ G1 G2 G3 ];
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Troubleshooting with Logs on page 323

Troubleshooting with Logs

Purpose Use these logs to identify any issues.

Action From operational mode, enter the **show log messages** command and the **show log dcd** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Client Groups for Firewall Authentication on page 321

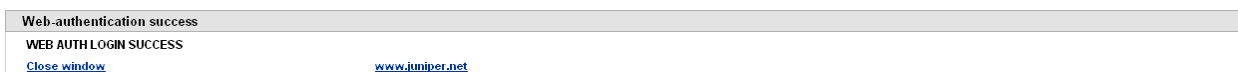
Firewall Authentication Banner Customization

- Understanding Firewall Authentication Banner Customization on page 323
- Example: Customizing a Firewall Authentication Banner on page 324

Understanding Firewall Authentication Banner Customization

A banner is a message that appears on a monitor in different places depending on the type of login. (See Figure 37 on page 323.)

Figure 37: Banner Customization



- At the top of a browser screen after a user has successfully logged into a Web authentication address as shown Figure 37 on page 323
- Before or after a Telnet, an FTP, or an HTTP login prompt, success message, and fail message for users

All of the banners, except for the one for a console login, already have default messages. You can customize the messages that appear on the banners to better suit the network environment in which you use the device.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Firewall User Authentication Overview on page 297
 - Example: Customizing a Firewall Authentication Banner on page 324

Example: Customizing a Firewall Authentication Banner

This example shows how to customize the banner text that appears in the browser.

- Requirements on page 324
- Overview on page 324
- Configuration on page 324
- Verification on page 325

Requirements

Before you begin, create an access profile.

Overview

A banner is a message that appears on a monitor in different places depending on the type of login. This example shows how to change the banner that appears in the browser to indicate that a user has successfully authenticated after successfully logging in through Web authentication. The new message is “Web authentication is successful.” If the authentication fails, then the new message reads “Authentication failed.”

Configuration

- CLI Quick Configuration** To quickly customize the banner text that appears in the browser, copy the following commands and paste them into the CLI:

```
[edit]
set access firewall-authentication pass-through default-profile Profile-1
set access firewall-authentication pass-through ftp banner fail " Authentication
failed"
set access firewall-authentication web-authentication default-profile Profile-1
set access firewall-authentication web-authentication banner success " Web
authentication is successful"
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To customize the banner text that appears in the browser:

1. Specify the banner text for failed pass-through authentication through FTP.

```
[edit]
user@host# set access firewall-authentication pass-through default-profile Profile-1
user@host# set access firewall-authentication pass-through ftp banner fail "
Authentication failed"
```
2. Specify the banner text for successful Web authentication.

```
[edit]
user@host# set access web-authentication default-profile Profile-1
user@host# set access web-authentication banner success " Web authentication
is successful"
```

Results From configuration mode, confirm your configuration by entering the **show access firewall-authentication** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show access firewall-authentication
pass-through {
  default-profile Profile-1;
  ftp {
    banner {
      fail "Authentication failed";
    }
  }
}
web-authentication {
  default-profile Profile-1;
  banner {
    success "Web authentication is successful";
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Troubleshooting with Logs on page 325

Troubleshooting with Logs

Purpose Use these logs to identify any issues.

Action From operational mode, enter the **show log messages** command and the **show log dcd** command.

Related Topics • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- [Understanding Firewall Authentication Banner Customization on page 323](#)

Infranet Authentication

- UAC and Junos OS on page 327
- Junos OS Enforcer and Infranet Controller Communications on page 329
- Junos OS Enforcer Policy Enforcement on page 332
- Junos OS Enforcer and IPsec on page 334
- Junos OS Enforcer and Infranet Agent Endpoint Security on page 342
- Junos OS Enforcer and Captive Portal on page 343
- Junos OS Enforcer and Infranet Controller Cluster Failover on page 351

UAC and Junos OS

- Understanding UAC in a Junos OS Environment on page 327
- Enabling UAC in a Junos OS Environment (CLI Procedure) on page 328

Understanding UAC in a Junos OS Environment

A Unified Access Control (UAC) deployment uses the following components to secure a network and ensure that only qualified end users can access protected resources:

- Infranet Controllers—An Infranet Controller is a policy decision point in the network. It uses authentication information and policy rules to determine whether or not to provide access to individual resources on the network. You can deploy one or more Infranet Controllers in your network.



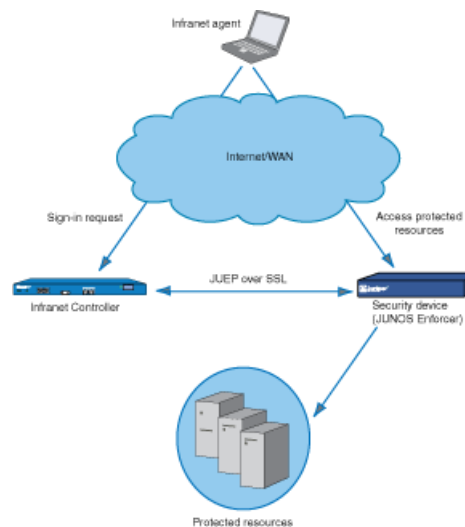
NOTE: Any change in the Unified Access Control's (UAC) contact interval and timeout values in the SRX Series or J Series device will be effective only after the next reconnection of the SRX Series or J Series device with the Infranet Controller.

- Infranet Enforcers—An Infranet Enforcer is a policy enforcement point in the network. It receives policies from the Infranet Controller and uses the rules defined in those policies to determine whether or not to allow an endpoint access to a resource. You deploy the Infranet Enforcers in front of the servers and resources that you want to protect.
- Infranet agents—An Infranet agent is a client-side component that runs directly on network endpoints (such as users' computers). The agent checks that the endpoint

complies to the security criteria specified in Host Checker policies and relays that compliance information to the Infranet Enforcer. The Infranet Enforcer then allows or denies the endpoint access based on the compliance results.

An SRX Series or J Series device can act as an Infranet Enforcer in a UAC network. Specifically, it acts as a Layer 3 enforcement point, controlling access by using IP-based policies pushed down from the Infranet Controller. When deployed in a UAC network, an SRX Series or J Series device is called a *Junos OS Enforcer*. See Figure 38 on page 328.

Figure 38: Integrating a Junos Security Device into a Unified Access Control Network



NOTE: You can use the Junos OS Enforcer with the Infranet Controller and Secure Access devices in an *IF-MAP Federation* network. In a federated network, multiple Infranet Controllers and Secure Access devices that are not directly connected to the Junos OS Enforcer can access resources protected by the security device. There are no configuration tasks for IF-MAP Federation on the Junos OS Enforcer. You configure policies on Infranet Controllers that can dynamically create authentication table entries on the Junos OS Enforcer. See the *Unified Access Control Administration Guide*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - *Unified Access Control Administration Guide*
 - Enabling UAC in a Junos OS Environment (CLI Procedure) on page 328

Enabling UAC in a Junos OS Environment (CLI Procedure)

Junos OS security policies enforce rules for transit traffic, defining what traffic can pass through the Juniper Networks device. The policies control traffic that enters from one zone (from-zone) and exits another (to-zone). To enable an SRX Series or J Series device as a Junos OS Enforcer in a UAC deployment, you must:

- Identify the source and destination zones through which UAC traffic will travel. It also needs the list of interfaces, including which zones they are in. The Infranet Controller uses the destination zone to match its own IPsec routing policies configured on Infranet Controller.
- Identify Junos OS security policies that encompass those zones, and enable UAC for those policies.

Before you begin:

1. Set up the interfaces through which UAC traffic should enter the SRX Series or J Series device. See *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Group interfaces with identical security requirements into zones. See “Example: Creating Security Zones” on page 88.
3. Create security policies to control the traffic that passes through the security zones. See “Example: Configuring a Security Policy to Permit or Deny All Traffic” on page 121.

To configure UAC through a Junos OS security policy, enter the following configuration statement:

```
user@host# set security policies from-zone zone-name to-zone zone-name policy match
then permit application-services uac-policy
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding UAC in a Junos OS Environment on page 327

Junos OS Enforcer and Infranet Controller Communications

- Understanding Communications Between the Junos OS Enforcer and the Infranet Controller on page 329
- Configuring Communications Between the Junos OS Enforcer and the Infranet Controller (CLI Procedure) on page 330

Understanding Communications Between the Junos OS Enforcer and the Infranet Controller

When you configure an SRX Series or J Series device to connect to an Infranet Controller, the SRX Series or J Series device and the Infranet Controller establish secure communications as follows:

1. The Infranet Controller presents its server certificate to the SRX Series or J Series device. If configured to do so, the SRX Series or J Series device verifies the certificate. (Server certificate verification is not required; however, as an extra security measure you can verify the certificate to implement an additional layer of trust.)
2. The SRX Series or J Series device and the Infranet Controller perform mutual authentication using the proprietary challenge-response authentication. For security reasons, the password is not included in the message sent to the Infranet Controller.
3. After successfully authenticating the SRX Series or J Series device, the Infranet Controller sends it user authentication and resource access policy information. The

SRX Series and J Series devices use this information to act as the Junos OS Enforcer in the UAC network.

4. Thereafter, the Infranet Controller and the Junos OS Enforcer can communicate freely with one another over the SSL connection. The communications are controlled by a proprietary protocol called *Junos UAC Enforcer Protocol (JUEP)*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - *Unified Access Control Administration Guide*
 - Understanding UAC in a Junos OS Environment on page 327
 - Configuring Communications Between the Junos OS Enforcer and the Infranet Controller (CLI Procedure) on page 330

Configuring Communications Between the Junos OS Enforcer and the Infranet Controller (CLI Procedure)

To configure an SRX Series or J Series device to act as a Junos OS Enforcer in a UAC deployment, and therefore to enforce Infranet Controller policies, you must specify an Infranet Controller to which the SRX Series or J Series device should connect.

Before you begin:

1. Enable UAC through the relevant Junos OS security policies. See “Enabling UAC in a Junos OS Environment (CLI Procedure)” on page 328.
2. (Optional) Import the Infranet Controller’s server certificate onto the SRX Series or J Series device and create a profile for the certificate authority (CA) that signed the certificate. See “Example: Loading CA and Local Certificates Manually (CLI)” on page 395.
3. Configure user authentication and authorization by setting up user roles, authentication and authorization servers, and authentication realms on the Infranet Controller. See the *Unified Access Control Administration Guide*.
4. Configure resource access policies on the Infranet Controller to specify which endpoints are allowed or denied access to protected resources. See the *Unified Access Control Administration Guide*.

To configure an SRX Series or J Series device to act as a Junos OS Enforcer:

1. Specify the Infranet Controller(s) to which the SRX Series or J Series device should connect.
 - To specify the Infranet Controller’s hostname:

```
user@host# set services unified-access-control infranet-controller hostname
```
 - To specify the Infranet Controller’s IP address:

```
user@host# set services unified-access-control infranet-controller hostname  
address ip-address
```



NOTE: When configuring access to multiple Infranet Controllers, you must define each separately. For example:

```
user@host# set services unified-access-control infranet-controller IC1
user@host# set services unified-access-control infranet-controller IC2
user@host# set services unified-access-control infranet-controller IC3

user@host# set services unified-access-control infranet-controller IC1 address 10.10.10.1
user@host# set services unified-access-control infranet-controller IC2 address 10.10.10.2
user@host# set services unified-access-control infranet-controller IC3 address 10.10.10.3
```

Make sure that all of the Infranet Controllers are members of the same cluster.



NOTE: By default, the Infranet Controller should select port 11123. To determine if this default has changed, see the *Unified Access Control Administration Guide*.

2. Specify the Junos OS interface to which the Infranet Controller should connect:

```
user@host# set services unified-access-control infranet-controller hostname interface
interface-name
```

3. Specify the password that the SRX Series or J Series device should use to initiate secure communications with the Infranet Controller:



NOTE: Any change in the Unified Access Control's (UAC) contact interval and timeout values in the SRX Series or J Series device will be effective only after the next reconnection of the SRX Series or J Series device with the Infranet Controller.

```
user@host# set services unified-access-control infranet-controller hostname
password password
```

4. (Optional) Specify information about the certificate that the device should use for SSL communications with the Infranet Controller.

- To specify the certificate that the device should use:

```
user@host# set services unified-access-control infranet-controller hostname
server-certificate-subject certificate-name
```

- To specify the CA profile associated with the certificate:

```
user@host# set services unified-access-control infranet-controller hostname
ca-profile ca-profile
```

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Communications Between the Junos OS Enforcer and the Infranet Controller on page 329

Junos OS Enforcer Policy Enforcement

- Understanding Junos OS Enforcer Policy Enforcement on page 332
- Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure) on page 333
- Verifying Junos OS Enforcer Policy Enforcement on page 334

Understanding Junos OS Enforcer Policy Enforcement

Once the SRX Series or J Series device has successfully established itself as the Junos OS Enforcer, it secures traffic as follows:

1. First, the Junos OS Enforcer uses the appropriate Junos OS security policy to process the traffic. A *security policy* uses criteria such as the traffic's source IP address or the time of day that the traffic was received to determine whether or not the traffic should be allowed to pass.
2. Once it determines that the traffic may pass based on the Junos OS security policy, the Junos OS Enforcer maps the traffic flow to an authentication table entry. The Junos OS Enforcer uses the source IP address of the first packet in the flow to create the mapping.

An *authentication table entry* contains the source IP address and user role(s) of a user who has already successfully established a UAC session. A *user role* identifies a group of users based on criteria such as type (for instance, "Engineering" or "Marketing") or status (for instance, "Antivirus Running"). The Junos OS Enforcer determines whether to allow or deny the traffic to pass based on the authentication results stored in the appropriate authentication table entry.

The Infranet Controller pushes authentication table entries to the Junos OS Enforcer when the devices first connect to one another and, as necessary, throughout the session. For example, the Infranet Controller might push updated authentication table entries to the Junos OS Enforcer when the user's computer becomes noncompliant with endpoint security policies, when you change the configuration of a user's role, or when you disable all user accounts on the Infranet Controller in response to a security problem such as a virus on the network.

If the Junos OS Enforcer drops a packet due to a missing authentication table entry, the device sends a message to the Infranet Controller, which in turn may provision a new authentication table entry and send it to the Junos OS Enforcer. This process is called *dynamic authentication table provisioning*.

3. Once it determines that the traffic may pass based on the authentication table entries, the Junos OS Enforcer maps the flow to a resource. The Junos OS Enforcer uses the destination IP address specified in the flow to create the mapping. Then the device uses that resource as well as the user role specified in the authentication table entry to map the flow to a resource access policy.

A *resource access policy* specifies a particular resource to which you want to control access based on user role. For instance, you might create a resource access policy that allows only users who are members of the Engineering and Antivirus Running

user roles access to the Engineering-Only server. Or you might create a resource access policy that allows members of the No Antivirus Running user role access to the Remediation server on which antivirus software is available for download.

The Infranet Controller pushes resource access policies to the Junos OS Enforcer when the devices first connect to one another and when you modify your resource access policy configurations on the Infranet Controller.

If the Junos OS Enforcer drops the packet because of a “deny” policy, the Junos OS Enforcer sends a message to the Infranet Controller, which in turn sends a message to the endpoint’s Odyssey Access Client (if available). (The Infranet Controller does not send “deny” messages to the agentless client.)

4. Once it determines that the traffic may pass based on the resource access policies, the Junos OS Enforcer processes the traffic using the remaining application services defined in the Junos OS policy. The Junos OS Enforcer runs the remaining services in the following order: Intrusion Detection and Prevention (IDP), URL filtering, and Application Layer Gateways (ALGs).

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- *Unified Access Control Administration Guide*
- Understanding Communications Between the Junos OS Enforcer and the Infranet Controller on page 329
- Security Policies Overview on page 115
- Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure) on page 333
- Verifying Junos OS Enforcer Policy Enforcement on page 334

Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure)

When configured in test-only mode, the SRX Series or J Series device enables all UAC traffic to go through regardless of the UAC policy settings. The device logs the UAC policy’s access decisions without enforcing them so you can test the implementation without impeding traffic.

Before you begin:

1. Enable UAC through the relevant Junos OS security policies. See “Enabling UAC in a Junos OS Environment (CLI Procedure)” on page 328
2. Configure the SRX Series and J Series devices as a Junos OS Enforcer. See “Configuring Communications Between the Junos OS Enforcer and the Infranet Controller (CLI Procedure)” on page 330.
3. If you are connecting to a cluster of Infranet Controllers, enable failover options. See “Configuring Junos OS Enforcer Failover Options (CLI Procedure)” on page 351.

To activate or deactivate test-only mode, enter the following configuration statement:

```
user@host# set services unified-access-control test-only-mode (true | false)
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Junos OS Enforcer Policy Enforcement on page 332
 - Verifying Junos OS Enforcer Policy Enforcement on page 334

Verifying Junos OS Enforcer Policy Enforcement

- Displaying Infranet Controller Authentication Table Entries from the Junos OS Enforcer on page 334
- Displaying Infranet Controller Resource Access Policies from the Junos OS Enforcer on page 334

Displaying Infranet Controller Authentication Table Entries from the Junos OS Enforcer

Purpose Display a summary of the authentication table entries configured from the Infranet Controller.

Action Enter the **show services unified-access-control authentication-table** CLI command.

Displaying Infranet Controller Resource Access Policies from the Junos OS Enforcer

Purpose Display a summary of UAC resource access policies configured from the Infranet Controller.

Action Enter the **show services unified-access-control policies** CLI command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Junos OS Enforcer Policy Enforcement on page 332
 - Testing Junos OS Enforcer Policy Access Decisions Using Test-Only Mode (CLI Procedure) on page 333
 - *Junos OS CLI Reference*

Junos OS Enforcer and IPsec

- Understanding Junos OS Enforcer Implementations Using IPsec on page 334
- Example: Configuring the Device as a Junos OS Enforcer Using IPsec (CLI) on page 336

Understanding Junos OS Enforcer Implementations Using IPsec

To configure an SRX Series or J Series device to act as a Junos OS Enforcer using IPsec, you must:

- Include the identity configured under the security IKE gateway. The identity is a string such as "gateway1.juniper.net", where gateway1.juniper.net distinguishes between IKE gateways. (The identities specify for which tunnel traffic is intended.)

- Include the preshared seed. This generates the preshared key from the full identity of the remote user for phase 1 credentials.
- Include the RADIUS shared secret. This allows the Infranet Controller to accept RADIUS packets for extended authentication (XAuth) from the Junos OS Infranet Enforcer.

When configuring IPsec between the Infranet Controller, the Odyssey Access Client, and the SRX or J Series device, you should note that the following are IKE (or phase 1) proposal methods or protocol configurations that are supported from the Infranet Controller to the Odyssey Access Client:

- IKE proposal: **authentication-method pre-shared-keys** (you must specify **pre-shared-keys**)
- IKE policy:
 - **mode aggressive** (you must use aggressive mode)
 - **pre-shared-key ascii-text key** (only ASCII text preshared-keys are supported)
- IKE gateway: dynamic
 - **hostname identity** (you must specify a unique identity among gateways)
 - **ike-user-type group-ike-id** (you must specify **group-ike-id**)
 - **xauth access-profile profile** (you must specify **xauth**)

The following are IPsec (or phase 2) proposal methods or protocol configurations that are supported from the Infranet Controller to the Odyssey Access Client.

- IPsec proposal: **protocol esp** (you must specify **esp**)
- IPsec VPN: **establish-tunnels immediately** (you must specify **establish-tunnels immediately**)



NOTE:

- Only one IPsec VPN tunnel is supported per from-zone to to-zone security policy. This is a limitation on the Infranet Controller.
 - Junos OS security policies enable you to define multiple policies differentiated by different source addresses, destination addresses, or both. The Infranet Controller, however, cannot differentiate such configurations. If you enable multiple policies in this manner, the Infranet Controller could potentially identify the incorrect IKE gateway.
-

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- *Unified Access Control Administration Guide*
- Understanding Junos OS Enforcer Policy Enforcement on page 332
- VPN Overview on page 355
- Security Policies Overview on page 115

- Example: Configuring the Device as a Junos OS Enforcer Using IPsec (CLI) on page 336

Example: Configuring the Device as a Junos OS Enforcer Using IPsec (CLI)

To configure an SRX Series or J Series device to act as a Junos OS Enforcer using IPsec:

1. Set system and syslog information using the following configuration statements:

```
system {
  host-name test_host;
  domain-name test.juniper.net;
  host-name test_host;
  root-authentication {
    encrypted-password "$1$uhqXoDOT$6h26f0xXExOqkPHQLvaTF0";
  }
  services {
    ftp;
    ssh;
    telnet;
    web-management {
      http {
        interface ge-0/0/0.0;
      }
    }
  }
}
syslog {
  user * {
    any emergency;
  }
  file messages {
    any critical;
    authorization info;
  }
  file interactive-commands {
    interactive-commands error;
  }
}
max-configurations-on-flash 5;
max-configuration-rollback 5;
license {
  autoupdate {
    url https://ae1.juniper.net/junos/key_retrieval;
  }
}
ntp {
  boot-server 1.2.3.4;
  server 1.2.3.4;
}
}
```

2. Configure the interfaces using the following configuration statements:

```
interfaces {
  ge-0/0/0 {
    unit 0 {
      family inet {
```



```

        address 10.64.75.135/16;
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.100.54.1/16;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 10.101.54.1/16;
        }
    }
}
}

```

3. Configure routing options using the following configuration statements:

```

routing-options {
    static {
        route 0.0.0.0/0 next-hop 10.64.0.1;
        route 10.11.0.0/16 next-hop 10.64.0.1;
        route 172.0.0.0/8 next-hop 10.64.0.1;
        route 10.64.0.0/16 next-hop 10.64.0.1;
    }
}

```

4. Configure security options using the following configuration statements:

```

security {
    ike {
        traceoptions {
            file ike;
            flag all;
        }
        proposal prop1 {
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
        }
        policy pol1 {
            mode aggressive;
            proposals prop1;
            pre-shared-key ascii-text "$9$YS4ZjmPQ6CuTz6Au0cSvWLxNbiHm";
        }
        gateway gateway1 {
            ike-policy pol1;
            dynamic {
                hostname gateway1.juniper.net;
                connections-limit 1000;
                ike-user-type group-ike-id;
            }
            external-interface ge-0/0/0;
        }
    }
}

```

```
xauth access-profile infranet;
}
gateway gateway2 {
ike-policy pol1;
  dynamic {
    hostname gateway2.juniper.net;
    connections-limit 1000;
    ike-user-type group-ike-id;
  }
  external-interface ge-0/0/0;
xauth access-profile infranet;
}
}
```

5. Configure IPsec parameters using the following configuration statements:

```
ipsec {
proposal prop1 {
  protocol esp;
  authentication-algorithm hmac-sha1-96;
  encryption-algorithm 3des-cbc;
  lifetime-seconds 86400;
}
policy pol1 {
proposals prop1;
}
vpn vpn1 {
ike {
  gateway gateway1;
  ipsec-policy pol1;
}
establish-tunnels immediately;
}
vpn vpn2 {
ike {
  gateway gateway2;
  ipsec-policy pol1;
}
establish-tunnels immediately;
}
}
```

6. Configure screen options using the following configuration statements:

```
screen {
ids-option untrust-screen {
icmp {
  ping-death;
}
ip {
  source-route-option;
  tear-drop;
}
tcp {
  syn-flood {
    alarm-threshold 1024;
    attack-threshold 200;
  }
}
}
```

```

        source-threshold 1024;
        destination-threshold 2048;
        queue-size 2000;
        timeout 20;
    }
    land;
}
}
}

```

7. Configure zones using the following configuration statements:

```

zones {
  security-zone trust {
    tcp-rst;
    host-inbound-traffic {
      system-services {
        all;
      }
      protocols {
        all;
      }
    }
    interfaces {
      ge-0/0/0.0;
    }
  }
  security-zone untrust {
    host-inbound-traffic {
      system-services {
        all;
      }
      protocols {
        all;
      }
    }
    interfaces {
      ge-0/0/1.0;
    }
  }
  security-zone zone101 {
    host-inbound-traffic {
      system-services {
        all;
      }
      protocols {
        all;
      }
    }
    interfaces {
      ge-0/0/2.0;
    }
  }
}
}

```

8. Configure policies for UAC using the following configuration statements:

```
policies {
  inactive: from-zone trust to-zone trust {
    policy default-permit {
      match {
        source-address any;
        destination-address any;
        application any;
      }
      then {
        permit;
      }
    }
  }
  from-zone trust to-zone untrust {
    inactive: policy default-permit {
      match {
        source-address any;
        destination-address any;
        application any;
      }
      then {
        permit;
      }
    }
    inactive: policy default-deny {
      match {
        source-address any;
        destination-address any;
        application any;
      }
      then {
        permit;
      }
    }
    policy pol1 {
      match {
        source-address any;
        destination-address any;
        application any;
      }
      then {
        permit {
          tunnel {
            ipsec-vpn vpn1;
          }
          application-services {
            uac-policy;
          }
        }
        log {
          session-init;
          session-close;
        }
      }
    }
  }
}
```

```
from-zone untrust to-zone trust {
policy pol1 {
  match {
    source-address any;
    destination-address any;
    application any;
  }
  then {
    permit;
    log {
      session-init;
      session-close;
    }
  }
}
}
from-zone trust to-zone zone101 {
policy pol1 {
  match {
    source-address any;
    destination-address any;
    application any;
  }
  then {
    permit {
      tunnel {
        ipsec-vpn vpn2;
      }
      application-services {
        uac-policy;
      }
    }
    log {
      session-init;
      session-close;
    }
  }
}
}
policy test {
match {
  source-address any;
  destination-address any;
  application any;
}
then {
  permit;
}
}
}
}
default-policy {
deny-all;
}
}
}
```

9. Configure RADIUS server authentication access using the following configuration statements:

```
access {
  profile infranet {
    authentication-order radius;
    radius-server {
      10.64.160.120 secret "$9$KBoWX-YgJHqfVwqfTzCAvWL";
    }
  }
}
```

10. Configure services for UAC using the following configuration statements:

```
services {
  unified-access-control {
    inactive: infranet-controller IC27 {
      address 3.23.1.2;
      interface ge-0/0/0.0;
      password "$9$Wjl8X-Vb2GDkev4aGUHkuOB";
    }
    infranet-controller prabaIC {
      address 10.64.160.120;
      interface ge-0/0/0.0;
      password "$9$jdkmT69pRhrz3hrev7Nik.";
    }
    traceoptions {
      flag all;
    }
  }
}
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Junos OS Enforcer Implementations Using IPsec on page 334

Junos OS Enforcer and Infranet Agent Endpoint Security

- Understanding Endpoint Security Using the Infranet Agent with the Junos OS Enforcer on page 342
- Configuring Endpoint Security Using the Infranet Agent with the Junos OS Enforcer on page 343

Understanding Endpoint Security Using the Infranet Agent with the Junos OS Enforcer

An Infranet agent helps you secure traffic on your network starting with the endpoints that initiate communications as follows:

1. The Infranet agent, which runs directly on the endpoint, checks that the endpoint is compliant with your Unified Access Control (UAC) Host Checker policies. You can use a wide variety of criteria within a UAC Host Checker policy to determine compliance. For example, you can configure the Host Checker policy to confirm that the endpoint is running antivirus software or a firewall or that the endpoint is not running specific types of malware or processes.

2. The Infranet agent transmits the compliance information to the Junos OS Enforcer.
3. The Junos OS Enforcer allows or denies the endpoint access to the resources on your network based on the Host Checker compliance results.

Because the Infranet agent runs directly on the endpoint, you can use the Infranet agent to check the endpoint for security compliance at any time. For instance, when a user tries to sign into the Infranet Controller, you can require the Infranet agent to send compliance results immediately—the user will not even see the sign-in page until the Infranet agent returns positive compliance results to the Infranet Controller. You can also configure the Infranet agent to check for compliance after the user signs in or periodically during the user session.

If the endpoints running the Infranet agent have appropriate access, they will automatically send their compliance results to the Infranet Controller and the Infranet Controller will update the authentication table entries accordingly and push them to the Junos OS Enforcer. The Junos OS Enforcer supports connections with the Odyssey Access Client and “agentless” Infranet agents.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - *Unified Access Control Administration Guide*
 - Understanding UAC in a Junos OS Environment on page 327
 - Configuring Endpoint Security Using the Infranet Agent with the Junos OS Enforcer on page 343

Configuring Endpoint Security Using the Infranet Agent with the Junos OS Enforcer

To integrate the Infranet agent into a Junos OS-UAC deployment, no special configuration is required on the Junos OS Enforcer. You simply need to create security policies enabling access to the appropriate endpoints as you would for any other Junos OS-UAC deployment.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Understanding Endpoint Security Using the Infranet Agent with the Junos OS Enforcer on page 342

Junos OS Enforcer and Captive Portal

- Understanding the Captive Portal on the Junos OS Enforcer on page 344
- Understanding Captive Portal Configuration on the Junos OS Enforcer on page 345
- Example: Creating a Captive Portal Policy on the Junos OS Enforcer (CLI) on page 346
- Understanding the Captive Portal Redirect URL Options on page 348
- Example: Configuring a Redirect URL for Captive Portal (CLI) on page 349

Understanding the Captive Portal on the Junos OS Enforcer

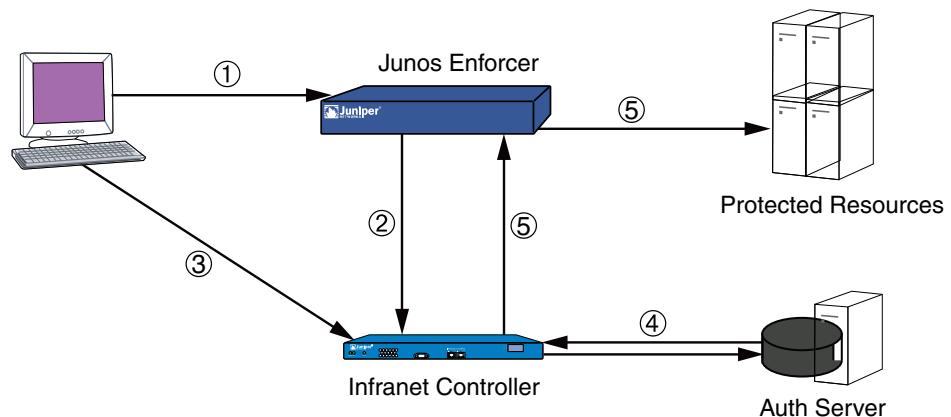
In a Unified Access Control (UAC) deployment, users might not be aware that they must first sign in to the Infranet Controller for authentication and endpoint security checking before they are allowed to access a protected resource behind the Junos OS Enforcers. To help users sign in to the Infranet Controller, you can configure the captive portal feature. The captive portal feature allows you to configure a policy in the Junos OS Enforcer that automatically redirects HTTP traffic destined for protected resources to the Infranet Controller or to a URL configured in the Junos OS Enforcer.

You can configure a captive portal for deployments that use either source IP enforcement or IPsec enforcement, or a combination of both enforcement methods.

Figure 39 on page 344 shows the captive portal feature enabled on a Junos OS Enforcer. Users accessing protected resources are automatically redirected to the Infranet Controller:

1. Users point to a protected resource using the browser.
2. The Junos OS Enforcer determines that the user is not authenticated and redirects the request to the Infranet Controller or another server.
3. Users enter their Infranet username and password to log in.
4. The Infranet Controller passes the user credentials to an authentication server.
5. After authentication, the Infranet Controller redirects the users to the protected resource they wanted to access.

Figure 39: Enabling the Captive Portal Feature on a Junos OS Enforcer



By default, the Junos OS Enforcer encodes and forwards to the Infranet Controller the protected resource URL that the user entered. The Infranet Controller uses the protected resource URL to help users navigate to the protected resource. The manner in which the Infranet Controller uses the protected resource URL depends on whether or not the user's endpoint is running the Odyssey Access Client or Junos Pulse. If the user's endpoint is not running the Odyssey Access Client or Junos Pulse (that is, it is in an agentless or Java agent configuration), the Infranet Controller automatically opens a new browser window and uses HTTP to access the protected resource after the user signs in. If the endpoint

is using the Odyssey Access Client, the Infranet Controller inserts a hypertext link in the webpage that automatically opens after the user signs in. The user must then click that hypertext link to access the protected resource by means of HTTP in the same browser window.

The Junos OS Enforcer supports the captive portal feature only for HTTP traffic. If you attempt to access a protected resource by using HTTPS or a non-browser application (such as an e-mail application), the Junos OS Enforcer does not redirect the traffic. When using HTTPS or a non-browser application, you must manually sign in to the Infranet Controller first before attempting to access protected resources.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding UAC in a Junos OS Environment on page 327
 - Understanding Captive Portal Configuration on the Junos OS Enforcer on page 345
 - Example: Creating a Captive Portal Policy on the Junos OS Enforcer (CLI) on page 346
 - Understanding the Captive Portal Redirect URL Options on page 348
 - Example: Configuring a Redirect URL for Captive Portal (CLI) on page 349

Understanding Captive Portal Configuration on the Junos OS Enforcer

To configure the captive portal feature, you create a security policy on the Junos OS Enforcer and then specify a redirection option for the captive portal security policy. You can choose to redirect traffic to an external server or to the Infranet Controller. You can also choose to redirect all traffic or unauthenticated traffic only.

- Redirecting traffic to an external webserver—You can configure the Junos OS Enforcer to redirect HTTP traffic to an external webserver instead of the Infranet Controller. For example, you can redirect HTTP traffic to a webpage that explains to users the requirement to sign in to the Infranet Controller before they can access the protected resource. You could also include a link to the Infranet Controller on that webpage to help users sign in.
- Redirecting unauthenticated traffic—Select this option if your deployment uses source IP only or a combination of source IP and IPsec. The Junos OS Enforcer redirects clear-text traffic from unauthenticated users to the currently connected Infranet Controller or to an IP address or domain name that you specify in a redirect URL. After a user signs in to the Infranet Controller and the user's endpoint system meets the requirements of the Infranet Controller's security policies, the Junos OS Enforcer allows the user's clear-text traffic to pass through in source IP deployments. For IPsec deployments, the Odyssey Access Client creates a VPN tunnel between the user and the Junos OS Enforcer. The Junos OS Enforcer then applies the VPN policy, allowing the encrypted traffic to pass through.
- Redirecting all traffic—Specify this option if you want to redirect all traffic to the URL that you specify in a redirect URL.
- Redirecting traffic with multiple Infranet Controllers—You can configure multiple Infranet Controllers on your Junos OS Enforcer, but it is connected to only one Infranet Controller at any given time. If the connection to the Infranet Controller fails, the Junos

OS Enforcer tries to connect to next configured Infranet Controller. As a result, you cannot be sure which Infranet Controller is connected to the Junos OS Enforcer at any given time. To ensure that the Junos OS Enforcer redirects traffic to the connected Infranet Controller configure the default redirect URL or the `%ic-ip%` option in the URL.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding UAC in a Junos OS Environment on page 327
 - Understanding the Captive Portal on the Junos OS Enforcer on page 344
 - Example: Creating a Captive Portal Policy on the Junos OS Enforcer (CLI) on page 346
 - Understanding the Captive Portal Redirect URL Options on page 348
 - Example: Configuring a Redirect URL for Captive Portal (CLI) on page 349

Example: Creating a Captive Portal Policy on the Junos OS Enforcer (CLI)

To configure the captive portal feature, you must create a captive portal policy. This example shows a simple configuration to illustrate the basic steps for creating a captive portal policy on the Junos OS Enforcer. In this example, you deploy a Junos OS Enforcer in front of the data center resources you want to protect and configure the captive portal feature on the Junos OS Enforcer. The Junos OS Enforcer then automatically redirects HTTP traffic destined for the protected resource to the Infranet Controller for authentication.

Before you configure the captive portal feature, be sure you have performed the following steps:

- Deploy the Infranet Controller in the network so that users can access the device. Use the internal port on the Infranet Controller to connect users, the Junos OS Enforcer (an SRX210 device in this example), and authentication servers. For instructions on how to configure the Infranet Controller, see “Configuring Communications Between the Junos OS Enforcer and the Infranet Controller (CLI Procedure)” on page 330.
- Set up security zones and interfaces on the Junos OS Enforcer. Make sure that end users are in a different security zone than protected resources. For example, protected resources in the data center are configured in the trusted zone and users in an untrusted zone.
- Add individual users to either an external authentication server or the local authentication server. Set up roles and realms for individual users. You can provision access to protected resources based on your network security needs. For more information about authentication tables and user roles, see the *Unified Access Control Administration Guide*.

In this example, you want to protect the trusted zone from users on the LAN by making sure that only compliant and authenticated users are granted access. New users join your network every month. You want to configure the captive portal feature on your system so that unauthenticated users are redirected to the Infranet Controller automatically without requiring new users to remember to log in to the Infranet Controller.

To configure the captive portal feature on the Junos OS Enforcer:

1. Create a security policy to be associated with the captive portal policy.

```
[edit]
user@host# edit security policies from-zone untrust to-zone trust policy my-policy
```
2. Specify the match condition for **my-policy**.

```
[edit security policies from-zone untrust to-zone trust policy my-policy]
user@host# set match destination-address any source-address any application any
```
3. Specify the captive portal policy as part of the UAC policy to be applied on the traffic that matches the specified conditions.

```
[edit security policies from-zone untrust to-zone trust policy my-policy]
user@host# set then permit application-services uac-policy captive-portal
my-captive-portal-policy
```
4. Navigate to the **services unified-access-control** level of the configuration hierarchy.

```
[edit security policies from-zone untrust to-zone trust policy my-policy]
user@host# top
[edit]
user@host# edit services unified-access-control
```
5. Specify to redirect all unauthenticated traffic to the Infranet Controller.

```
[edit services unified-access-control]
user@host# set captive-portal my-captive-portal-policy redirect-traffic
unauthenticated
```
6. Navigate to the top level of the configuration hierarchy.

```
[edit services unified-access-control]
user@host# top
```
7. Confirm your configuration by entering the **show services** and the **show security policies** commands from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this show output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show services
unified-access-control {
  captive-portal my-captive-portal-policy {
    redirect-traffic unauthenticated;
  }
}

[edit]
user@host# show security policies
...
from-zone untrust to-zone trust {
  policy my-policy {
    match {
      source-address any;
      destination-address any;
```

```

        application any;
    }
    then {
        permit {
            application-services {
                uac-policy {
                    captive-portal my-captive-portal-policy;
                }
            }
        }
    }
}

```

8. Commit the configuration if you are done configuring the device.

```

[edit]
user@host# commit

```

For more information about the configuration statements used in this example, see the *Junos OS CLI Reference*.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding UAC in a Junos OS Environment on page 327
- Understanding the Captive Portal on the Junos OS Enforcer on page 344
- Understanding Captive Portal Configuration on the Junos OS Enforcer on page 345
- Understanding the Captive Portal Redirect URL Options on page 348
- Example: Configuring a Redirect URL for Captive Portal (CLI) on page 349

Understanding the Captive Portal Redirect URL Options

By default, after you configure a captive portal policy, the Junos OS Enforcer redirects HTTP traffic to the currently connected Infranet Controller by using HTTPS. To perform the redirection, the Junos OS Enforcer uses the IP address or domain name that you specified when you configured the Infranet Controller instance on the Junos OS Enforcer. The format of the URL that the Junos OS Enforcer uses for default redirection is:

```
https://%ic-ip%/?target = %dest-url% &enforcer = %enforcer-id% &policy = %policy-id% &dest-ip = %dest-ip%
```

If you configured your Junos OS Enforcer to work with multiple Infranet Controllers in a cluster, and the current Infranet Controller becomes disconnected, the Junos OS Enforcer automatically redirects HTTP traffic to the next active Infranet Controller in its configuration list. The Junos OS Enforcer redirects traffic to only one Infranet Controller at a time.

Otherwise, the browser displays a certificate warning to users when they sign in. You do not need to override the default redirection destination except in these situations:

- You are using a VIP for a cluster of Infranet Controller appliances and the Junos OS Enforcer is configured to connect to the Infranet Controller's physical IP addresses.

- You want to redirect traffic to a webserver instead of the Infranet Controller.
- If, because of split DNS or IP routing restrictions at your site, the Junos OS Enforcer uses a different address for the Infranet Controller than endpoints, you must specify the domain name or IP address that endpoints must use to access the Infranet Controller.

Table 37 on page 349 lists different options that you can configure in the redirect URL string.

Table 37: Redirect URL String Options

%dest-url%	Specifies the protected resource which the user is trying to access.
%enforcer-id%	Specifies the ID assigned to the Junos OS Enforcer by the Infranet Controller.
%policy-id%	Specifies the encrypted policy ID for the captive portal security policy that redirected the traffic.
%dest-ip%	Specifies the IP address or hostname of the protected resource which the user is trying to access.
%ic-ip%	Specifies the IP address or hostname of the Infranet Controller to which the Junos OS Enforcer is currently connected to.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding UAC in a Junos OS Environment on page 327
 - Understanding the Captive Portal on the Junos OS Enforcer on page 344
 - Understanding Captive Portal Configuration on the Junos OS Enforcer on page 345
 - Example: Creating a Captive Portal Policy on the Junos OS Enforcer (CLI) on page 346
 - Example: Configuring a Redirect URL for Captive Portal (CLI) on page 349

Example: Configuring a Redirect URL for Captive Portal (CLI)

You can redirect traffic to the currently connected Infranet Controller or to an IP address or domain name that you specify in a redirect URL. We recommend the default configuration that redirects traffic to the Infranet Controller for authentication.

If you need to override the default redirection destination, you can specify any combination of redirect options:

- **https://<IP or domain name>/<URL path>/target=%dest-url%**—Configure this option to forward users to the protected resource automatically after authentication with the Infranet Controller or webserver. The Junos OS Enforcer replaces the **%dest-url%** parameter with the protected resource URL and then forwards the protected resource URL in encrypted form to the Infranet Controller.
- **https://<IP or domain name>/<URL path>**—Configure this option for users to be redirected to the Infranet Controller authentication page but not be forwarded to the

protected resource after authentication. Users must manually open a new browser window and enter the protected resource URL again after signing in.

- **redirect-all**—Specify this option if you want to redirect all traffic to the URL that you specify in a redirect URL.

In this example, you configure the URL to redirect traffic to the Infranet Controller and after authentication forward the traffic automatically to the protected resource.

Before you specify the redirect URL, make sure you configure the captive portal policy. For information about creating the captive portal policy, see “Example: Creating a Captive Portal Policy on the Junos OS Enforcer (CLI)” on page 346.

To configure the redirect URL for the captive portal feature on the Junos OS Enforcer:

1. Navigate to the **services unified-access-control** level of the configuration hierarchy.

```
[edit]
user@host# edit services unified-access-control
```

2. Specify the redirect URL for the preconfigured captive portal policy.

```
[edit services unified-access-control]
user@host# set captive-portal my-captive-portal-policy redirect-url
https://www.my-website.com
```

3. Navigate to the top level of the configuration hierarchy.

```
[edit services unified-access-control]
user@host# top
```

4. Confirm your configuration by entering the **show services** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show services
unified-access-control {
  captive-portal my-captive-portal-policy {
    redirect-url https://my-website.com;
  }
}
```

5. Commit the configuration if you are done configuring the device.

```
[edit]
user@host# commit
```

For more information about the configuration statements used in this example, see the *Junos OS CLI Reference*.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding UAC in a Junos OS Environment on page 327
- Understanding the Captive Portal on the Junos OS Enforcer on page 344
- Understanding Captive Portal Configuration on the Junos OS Enforcer on page 345

- Understanding the Captive Portal Redirect URL Options on page 348

Junos OS Enforcer and Infranet Controller Cluster Failover

- Understanding Communications Between Junos OS Enforcer and a Cluster of Infranet Controllers on page 351
- Configuring Junos OS Enforcer Failover Options (CLI Procedure) on page 351

Understanding Communications Between Junos OS Enforcer and a Cluster of Infranet Controllers

You can configure a Junos OS Enforcer to work with more than one Infranet Controller in a high availability configuration known as an Infranet Controller cluster. The Junos OS Enforcer communicates with only one Infranet Controller at a time; the other Infranet Controllers are used for failover. If the Junos OS Enforcer cannot connect to the first Infranet Controller you added to a cluster, it tries to connect to the failed Infranet Controller again. Then it fails over to the other Infranet Controllers in the cluster. It continues trying to connect to Infranet Controllers in the cluster until a connection occurs.

When the Junos OS Enforcer cannot establish a connection to an Infranet Enforcer, it preserves all its existing authentication table entries and Unified Access Control (UAC) policies and takes the timeout action that you specify. Timeout actions include:

- **close**—Close existing sessions and block any further traffic. This is the default option.
- **no-change**—Preserve existing sessions and require authentication for new sessions.
- **open**—Preserve existing sessions and allow new sessions access.

Once the Junos OS Enforcer can reestablish a connection to an Infranet Controller, the Infranet Controller compares the authentication table entries and UAC policies stored on the Junos OS Enforcer with the authentication table entries and policies stored on the Infranet Controller and reconciles the two as required.



NOTE: The Infranet Controllers configured on a Junos OS Enforcer should all be members of the same Infranet Controller cluster.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- *Unified Access Control Administration Guide*
- Understanding Junos OS Enforcer Policy Enforcement on page 332
- Understanding Junos OS Enforcer Policy Enforcement on page 332
- Configuring Junos OS Enforcer Failover Options (CLI Procedure) on page 351

Configuring Junos OS Enforcer Failover Options (CLI Procedure)

To configure Infranet Controller failover processing, you must configure the Junos OS Enforcer to connect to a cluster of Infranet Controllers. The Junos OS Enforcer

communicates with one of these Infranet Controllers at a time and uses the others for failover processing.

Before you begin:

1. Enable UAC through the relevant Junos OS security policies. See “Enabling UAC in a Junos OS Environment (CLI Procedure)” on page 328.
2. Configure the SRX Series or J Series device as a Junos OS Enforcer. During the configuration, define a cluster of Infranet Controllers to which the Junos OS Enforcer should connect. See “Configuring Communications Between the Junos OS Enforcer and the Infranet Controller (CLI Procedure)” on page 330.

To configure failover processing:

1. Specify how often (in seconds) the Junos OS Enforcer should expect a heartbeat signal from the Infranet Controller indicating an active connection:

```
user@host# set services unified-access-control interval seconds
```

2. Specify the interval (in seconds) at which the Junos OS Enforcer should consider the current connection timed out:

```
user@host# set services unified-access-control timeout seconds
```

3. Specify how the Junos OS Enforcer should handle all current and subsequent traffic sessions when its connection to an Infranet Controller cluster times out:

```
user@host# set services unified-access-control timeout-action (close | no-change | open)
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Communications Between Junos OS Enforcer and a Cluster of Infranet Controllers on page 351

PART 6

Virtual Private Networks

- Internet Protocol Security on page 355
- Public Key Cryptography for Certificates on page 383
- Dynamic VPNs on page 405
- Group VPNs on page 423

Internet Protocol Security

- VPN Overview on page 355
- Understanding IKE and IPsec Packet Processing on page 361
- IPsec VPN Configuration Overview on page 366
- Phase 1 Proposals for IPsec VPNs on page 367
- Phase 2 Proposals for IPsec VPNs on page 371
- Global SPI and VPN Monitoring Features on page 374
- Hub-and-Spoke VPNs on page 375

VPN Overview

A virtual private network (VPN) provides a means for securely communicating among remote computers across a public WAN such as the Internet.

A VPN connection can link two LANs (site-to-site VPN) or a remote dial-up user and a LAN. The traffic that flows between these two points passes through shared resources such as routers, switches, and other network equipment that make up the public WAN. To secure VPN communication while passing through the WAN, the two participants create an IP Security (IPsec) tunnel.



NOTE: The term *tunnel* does not denote tunnel mode (see “Packet Processing in Tunnel Mode” on page 361). Instead, it refers to the IPsec connection.

IPsec is a suite of related protocols for cryptographically securing communications at the IP Packet Layer. IPsec also provides methods for the manual and automatic negotiation of security associations (SAs) and key distribution, all the attributes for which are gathered in a Domain of Interpretation (DOI). The IPsec DOI is a document containing definitions for all the security parameters required for the successful negotiation of a VPN tunnel—essentially, all the attributes required for SA and IKE negotiations. See RFC 2407 and RFC 2408 for more information.

This topic includes the following sections:

- Security Associations on page 356
- IPsec Key Management on page 357

- IPsec Security Protocols on page 358
- IPsec Tunnel Negotiation on page 360
- Distributed VPNs in SRX Series Services Gateways on page 360

Security Associations

A security association (SA) is a unidirectional agreement between the VPN participants regarding the methods and parameters to use in securing a communication channel. Full bidirectional communication requires at least two SAs, one for each direction. Through the SA, an IPsec tunnel can provide the following security functions:

- Privacy (through encryption)
- Content integrity (through data authentication)
- Sender authentication and—if using certificates—nonrepudiation (through data origin authentication)

The security functions you employ depend on your needs. If you need only to authenticate the IP packet source and content integrity, you can authenticate the packet without applying any encryption. On the other hand, if you are concerned only with preserving privacy, you can encrypt the packet without applying any authentication mechanisms. Optionally, you can both encrypt and authenticate the packet. Most network security designers choose to encrypt, authenticate, and replay-protect their VPN traffic.

An IPsec tunnel consists of a pair of unidirectional SAs—one at each end of the tunnel—that specify the security parameter index (SPI), destination IP address, and security protocol (Authentication Header or Encapsulating Security Payload) employed. An SA groups together the following components for securing communications:

- Security algorithms and keys.
- Protocol mode, either transport or tunnel. Junos OS devices always use tunnel mode. (For more information, see “Packet Processing in Tunnel Mode” on page 361.)
- Key-management method, either manual key or AutoKey IKE. (For more information, see “IPsec Key Management” on page 357.)
- SA lifetime.

For inbound traffic, Junos OS looks up the SA by using the following triplet:

- Destination IP address.
- Security protocol, either AH or ESP. (See “IPsec Security Protocols” on page 358.)
- Security parameter index (SPI) value.

For outbound VPN traffic, the policy invokes the SA associated with the VPN tunnel.

IPsec Key Management

The distribution and management of keys are critical to using VPNs successfully. Junos OS supports IPsec technology for creating VPN tunnels with three kinds of key creation mechanisms:

- Manual key
- AutoKey IKE with a preshared key or a certificate
- Diffie-Hellman (DH) key

You can choose your key creation mechanism—also called authentication method—during Phase 1 and Phase 2 proposal configuration. For more information, see “IPsec Tunnel Negotiation” on page 360.



NOTE: Manual key creation and AutoKey IKE with certificates are not supported with the dynamic VPN feature at this time.

This topic includes the following sections:

- Manual Key on page 357
- AutoKey IKE on page 357
- Diffie-Hellman Exchange on page 358

Manual Key

With manual keys, administrators at both ends of a tunnel configure all the security parameters. This is a viable technique for small, static networks where the distribution, maintenance, and tracking of keys are not difficult. However, safely distributing manual-key configurations across great distances poses security issues. Aside from passing the keys face-to-face, you cannot be completely sure that the keys have not been compromised while in transit. Also, whenever you want to change the key, you are faced with the same security issues as when you initially distributed it.

AutoKey IKE

When you need to create and manage numerous tunnels, you need a method that does not require you to configure every element manually. IPsec supports the automated generation and negotiation of keys and security associations using the Internet Key Exchange (IKE) protocol. Junos OS refers to such automated tunnel negotiation as AutoKey IKE and supports AutoKey IKE with preshared keys and AutoKey IKE with certificates.

- AutoKey IKE with preshared keys—Using AutoKey IKE with preshared keys to authenticate the participants in an IKE session, each side must configure and securely exchange the preshared key in advance. In this regard, the issue of secure key distribution is the same as that with manual keys. However, once distributed, an autokey, unlike a manual key, can automatically change its keys at predetermined intervals using the IKE protocol. Frequently changing keys greatly improves security, and automatically doing so greatly reduces key-management responsibilities. However, changing keys

increases traffic overhead; therefore, doing so too often can reduce data transmission efficiency.



NOTE: A preshared key is a key for both encryption and decryption, which both participants must have before initiating communication.

- AutoKey IKE with certificates—When using certificates to authenticate the participants during an AutoKey IKE negotiation, each side generates a public-private key pair and acquires a certificate. As long as the issuing certificate authority (CA) is trusted by both sides, the participants can retrieve the peer's public key and verify the peer's signature. There is no need to keep track of the keys and SAs; IKE does it automatically.

Diffie-Hellman Exchange

A Diffie-Hellman (DH) exchange allows the participants to produce a shared secret value. The strength of the technique is that it allows the participants to create the secret value over an unsecured medium without passing the secret value through the wire. There are five DH groups; Junos OS supports groups 1, 2, and 5. The size of the prime modulus used in each group's calculation differs as follows:

- DH Group 1—768-bit modulus
- DH Group 2—1024-bit modulus
- DH Group 5—1536-bit modulus



NOTE: The strength of DH Group 1 security has depreciated; therefore we do not recommend its use.

The larger the modulus, the more secure the generated key is considered to be; however, the larger the modulus, the longer the key-generation process takes. Because the modulus for each DH group is a different size, the participants must agree to use the same group.



NOTE: If you configure multiple (up to four) proposals for Phase 1 negotiations, use the same DH group in all proposals. The same guideline applies to multiple proposals for Phase 2 negotiations.

IPsec Security Protocols

IPsec uses two protocols to secure communications at the IP layer:

- Authentication Header (AH)—A security protocol for authenticating the source of an IP packet and verifying the integrity of its content
- Encapsulating Security Payload (ESP)—A security protocol for encrypting the entire IP packet (and authenticating its content)

You can choose your security protocols—also called authentication and encryption algorithms—during Phase 1 and Phase 2 proposal configuration. For more information, see “IPsec Tunnel Negotiation” on page 360.

This topic includes the following sections:

- AH Protocol on page 359
- ESP Protocol on page 359

AH Protocol

The Authentication Header (AH) protocol provides a means to verify the authenticity/integrity of the content and origin of a packet. You can authenticate the packet by the checksum calculated through a Hash Message Authentication Code (HMAC) using a secret key and either MD5 or SHA-1 hash functions.

- Message Digest 5 (MD5)—An algorithm that produces a 128-bit hash (also called a *digital signature* or *message digest*) from a message of arbitrary length and a 16-byte key. The resulting hash is used, like a fingerprint of the input, to verify content and source authenticity and integrity.
- Secure Hash Algorithm (SHA-1)—An algorithm that produces a 160-bit hash from a message of arbitrary length and a 20-byte key. It is generally regarded as more secure than MD5 because of the larger hashes it produces. Because the computational processing is done in the ASIC, the performance cost is negligible.



NOTE: For more information on MD5 hashing algorithms, see RFC 1321 and RFC 2403. For more information on SHA hashing algorithms, see RFC 2404. For more information on HMAC, see RFC 2104.

ESP Protocol

The Encapsulating Security Payload (ESP) protocol provides a means to ensure privacy (encryption) and source authentication and content integrity (authentication). ESP in tunnel mode encapsulates the entire IP packet (header and payload) and then appends a new IP header to the now-encrypted packet. This new IP header contains the destination address needed to route the protected data through the network. (For more information about tunnel mode, see “Packet Processing in Tunnel Mode” on page 361.)

With ESP, you can both encrypt and authenticate, encrypt only, or authenticate only. For encryption, you can choose one of the following encryption algorithms:

- Data Encryption Standard (DES)—A cryptographic block algorithm with a 56-bit key.
- Triple DES (3DES)—A more powerful version of DES in which the original DES algorithm is applied in three rounds, using a 168-bit key. DES provides a significant performance savings but is considered unacceptable for many classified or sensitive material transfers.
- Advanced Encryption Standard (AES)—An emerging encryption standard which, when adopted by Internet infrastructures worldwide, will offer greater interoperability with other devices. Junos OS supports AES with 128-bit, 192-bit, and 256-bit keys.

For authentication, you can use either the MD5 or the SHA-1 algorithm.



NOTE: Even though it is possible to select **NULL** for encryption, it has been demonstrated that IPsec might be vulnerable to attack under such circumstances. Therefore, we suggest that you choose an encryption algorithm for maximum security.

IPsec Tunnel Negotiation

To establish an AutoKey IKE IPsec tunnel, two phases of negotiation are required:

- In Phase 1, the participants establish a secure channel in which to negotiate the IPsec security associations (SAs).
- In Phase 2, the participants negotiate the IPsec SAs for encrypting and authenticating the ensuing exchanges of user data.

For a manual key IPsec tunnel, because all of the SA parameters have been previously defined, there is no need to negotiate which SAs to use. In essence, the tunnel has already been established. When traffic matches a policy using that manual key tunnel or when a route involves the tunnel, the Juniper Networks device simply encrypts and authenticates the data, as you determined, and forwards it to the destination gateway.

Distributed VPNs in SRX Series Services Gateways

In the SRX3000 and SRX5000 lines, the IKE provides tunnel management for IPsec and authenticates end entities. The IKE performs a Diffie-Hellman (DH) key exchange to generate an IPsec tunnel between network devices. The IPsec tunnels generated by IKE are used to encrypt, decrypt, and authenticate user traffic between the network devices at the IP layer.

The VPN is created by distributing the IKE and IPsec workload among the multiple Security Processing Units (SPUs) of the platform. The IKE workload is distributed based on a key generated from the IKE packet's 4 tuples (source IP address, destination IP addresses, and UDP ports). The workload is distributed by assigning anchoring SPUs logically and mapping the logical SPUs to physical SPU-based on the composition at that given time. This distribution prevents any change in the number and composition of SPUs in the device, which may happen due to hot swap or SPC failure. The SPU in a device communicates with the Routing Engine to create a distributed VPN.

In IPsec, the workload is distributed by the same algorithm that distributes the IKE. The Phase 2 SA for a given VPN tunnel termination points pair is exclusively owned by a particular SPU, and all IPsec packets belonging to this Phase 2 SA are forwarded to the anchoring SPU of that security association for IPsec processing.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Phase 1 of IKE Tunnel Negotiation on page 367
 - Understanding Phase 2 of IKE Tunnel Negotiation on page 371
 - Understanding IKE and IPsec Packet Processing on page 361
 - Understanding Hub-and-Spoke VPNs on page 375

- IPsec VPN Configuration Overview on page 366

Understanding IKE and IPsec Packet Processing

An IPsec VPN tunnel consists of tunnel setup and applied security. During tunnel setup, the peers establish security associations (SAs), which define the parameters for securing traffic between themselves. (For more information, see “VPN Overview” on page 355.) After the tunnel is established, IPsec protects the traffic sent between the two tunnel endpoints by applying the security parameters defined by the SAs during tunnel setup. Within the Junos OS implementation, IPsec is applied in tunnel mode, which supports the Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols.

This topic includes the following sections:

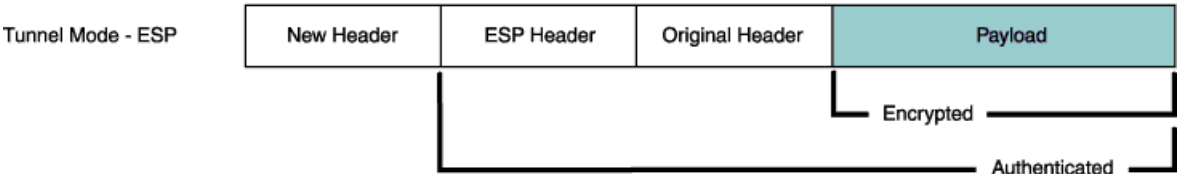
- Packet Processing in Tunnel Mode on page 361
- IKE Packet Processing on page 362
- IPsec Packet Processing on page 364

Packet Processing in Tunnel Mode

IPsec operates in one of two modes—transport or tunnel. When both ends of the tunnel are hosts, you can use either mode. When at least one of the endpoints of a tunnel is a security gateway, such as a Junos OS router or firewall, you must use tunnel mode. Juniper Networks devices always operate in tunnel mode for IPsec tunnels.

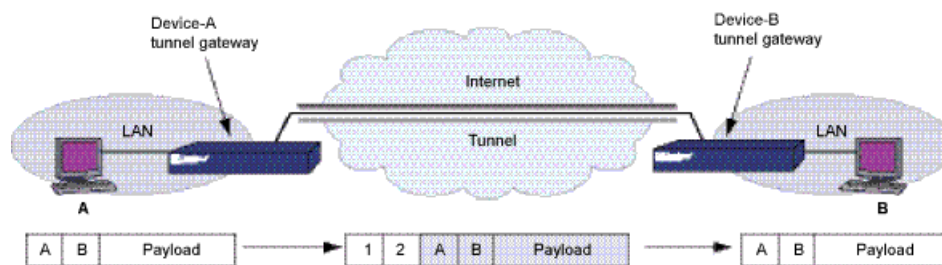
In tunnel mode, the entire original IP packet—payload and header—is encapsulated within another IP payload and a new header is appended to it, as shown in Figure 40 on page 361. The entire original packet can be encrypted, authenticated, or both. With the Authentication Header (AH) protocol, the AH and new headers are also authenticated. With the Encapsulating Security Payload (ESP) protocol, the ESP header can also be authenticated.

Figure 40: Tunnel Mode



In a site-to-site VPN, the source and destination addresses used in the new header are the IP addresses of the outgoing interface. See Figure 41 on page 362.

Figure 41: Site-to-Site VPN in Tunnel Mode

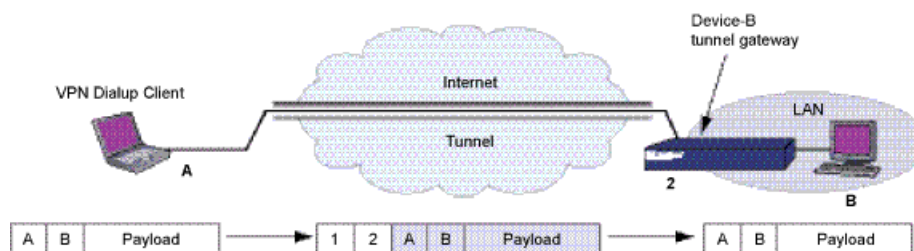


In a dial-up VPN, there is no tunnel gateway on the VPN dial-up client end of the tunnel; the tunnel extends directly to the client itself (see Figure 42 on page 362). In this case, on packets sent from the dial-up client, both the new header and the encapsulated original header have the same IP address: that of the client's computer.



NOTE: Some VPN clients such as the dynamic VPN client and Netscreen-Remote use a virtual inner IP address (also called a “sticky address”). Netscreen-Remote enables you to define the virtual IP address. The dynamic VPN client uses the virtual IP address assigned by the Radius server during the Xauth configuration exchange. In such cases, the virtual inner IP address is the source IP address in the original packet header of traffic originating from the client, and the IP address that the ISP dynamically assigns the dial-up client is the source IP address in the outer header.

Figure 42: Dial-Up VPN in Tunnel Mode

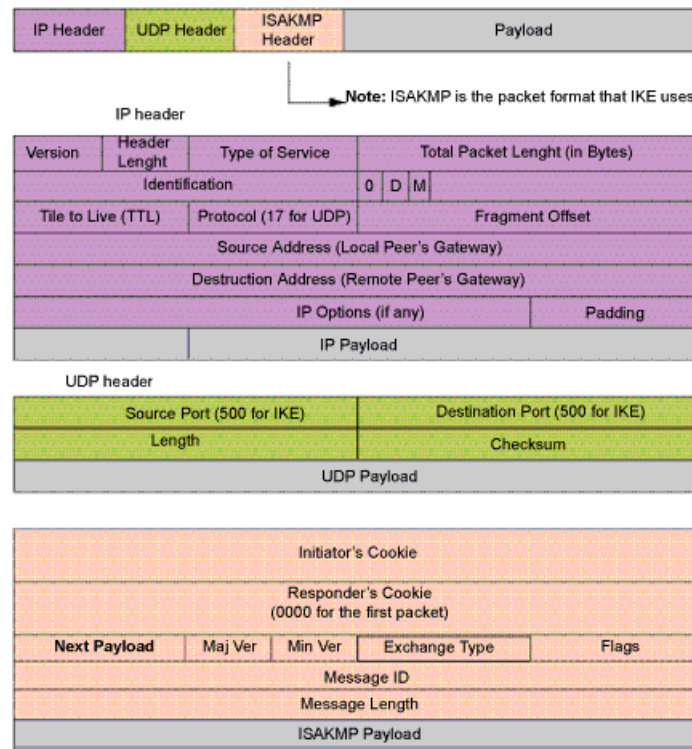


IKE Packet Processing

When a cleartext packet arrives on a Juniper Networks device that requires tunneling and no active Phase 2 SA exists for that tunnel, Junos OS begins IKE negotiations and drops the packet. The source and destination addresses in the IP packet header are those of the local and remote IKE gateways, respectively. In the IP packet payload, there is a UDP segment encapsulating an ISAKMP (IKE) packet. The format for IKE packets is the same for Phase 1 and Phase 2. See Figure 43 on page 363.

Meanwhile, the source host has sent the dropped packet again. Typically, by the time the second packet arrives, IKE negotiations are complete and Junos OS protects it—and all subsequent packets in the session—with IPsec before forwarding it.

Figure 43: IKE Packet for Phases 1 and 2



The Next Payload field contains a number indicating one of the following payload types:

- 0002—SA Negotiation Payload contains a definition for a Phase 1 or Phase 2 SA.
- 0004—Proposal Payload can be a Phase 1 or Phase 2 proposal.
- 0008—Transform Payload gets encapsulated in a proposal payload that gets encapsulated in an SA payload.
- 0010—Key Exchange (KE) Payload contains information necessary to perform a key exchange, such as a DH public value.
- 0020—Identification (IDx) Payload.
 - In Phase 1, IDii indicates the initiator ID, and IDir indicates the responder ID.
 - In Phase 2, IDui indicates the user initiator, and IDur indicates the user responder.

The IDs are IKE ID types such as FQDN, U-FQDN, IP address, and ASN.1_DN.

- 0040—Certificate (CERT) Payload.
- 0080—Certificate Request (CERT_REQ) Payload.
- 0100—Hash (HASH) Payload contains the digest output of a particular hash function.
- 0200—Signature (SIG) Payload contains a digital signature.
- 0400—Nonce (Nx) Payload contains some pseudorandom information necessary for the exchange).

- 0800—Notify Payload.
- 1000—ISAKMP Delete Payload.
- 2000—Vendor ID (VID) Payload can be included anywhere in Phase 1 negotiations. Junos OS uses it to mark support for NAT-T.

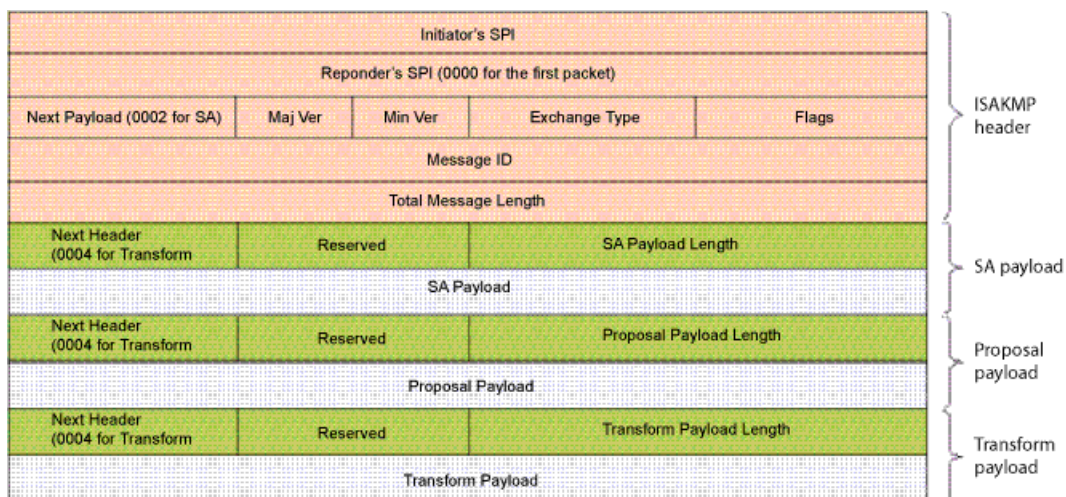
Each ISAKMP payload begins with the same generic header, as shown in Figure 44 on page 364.

Figure 44: Generic ISAKMP Payload Header

Next Header	Reserved	Payload Length (in bytes)
Payload		

There can be multiple ISAKMP payloads chained together, with each subsequent payload type indicated by the value in the Next Header field. A value of **0000** indicates the last ISAKMP payload. See Figure 45 on page 364 for an example.

Figure 45: ISAKMP Header with Generic ISAKMP Payloads



IPsec Packet Processing

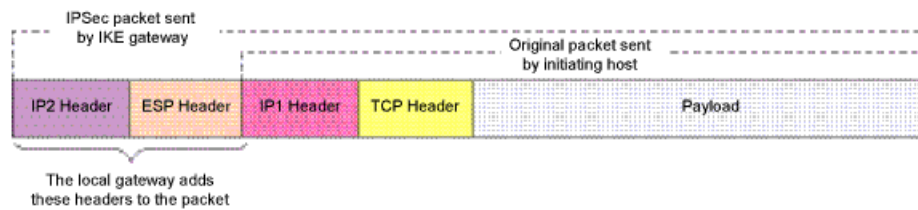
After IKE negotiations complete and the two IKE gateways have established Phase 1 and Phase 2 security associations (SAs), the device applies IPsec protection to subsequent cleartext IP packets that hosts behind one IKE gateway send to hosts behind the other gateway (assuming that policies permit the traffic). If the Phase 2 SA specifies the Encapsulating Security Protocol (ESP) in tunnel mode, the packet looks like the one shown below. The device adds two additional headers to the original packet that the initiating host sends.



NOTE: For information about ESP, see “ESP Protocol” on page 359. For information about tunnel mode, see “Packet Processing in Tunnel Mode” on page 361.

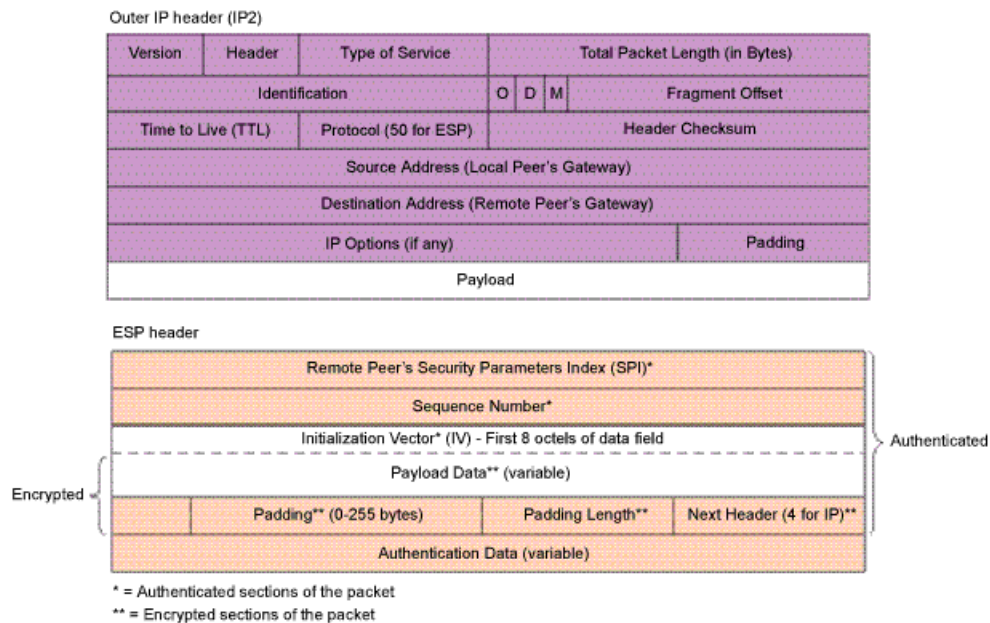
As shown in Figure 46 on page 365, the packet that the initiating host constructs includes the payload, the TCP header, and the inner IP header (IP1).

Figure 46: IPsec Packet—ESP in Tunnel Mode



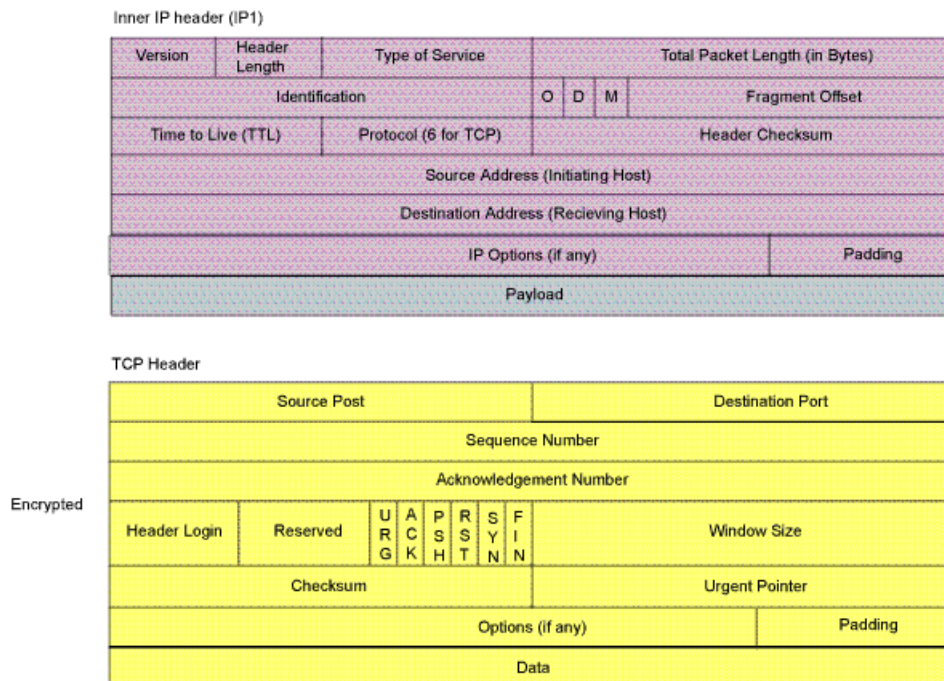
The router IP header (IP2), which Junos OS adds, contains the IP address of the remote gateway as the destination IP address and the IP address of the local router as the source IP address. Junos OS also adds an ESP header between the outer and inner IP headers. The ESP header contains information that allows the remote peer to properly process the packet when it receives it. This is illustrated in Figure 47 on page 365.

Figure 47: Outer IP Header (IP2) and ESP Header



The Next Header field indicates the type of data in the payload field. In tunnel mode, this value is 4, indicating IP-in-IP. See Figure 48 on page 366.

Figure 48: Inner IP Header (IP1) and TCP Header



- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - VPN Overview on page 355
 - Understanding Phase 1 of IKE Tunnel Negotiation on page 367
 - Understanding Phase 2 of IKE Tunnel Negotiation on page 371
 - Understanding Hub-and-Spoke VPNs on page 375
 - IPsec VPN Configuration Overview on page 366

IPsec VPN Configuration Overview

IKE IPsec tunnel negotiation occurs in two phases. In Phase 1, participants establish a secure channel in which to negotiate the IPsec security association (SA). In Phase 2, participants negotiate the IPsec SA for authenticating traffic that will flow through the tunnel. Just as there are two phases to tunnel negotiation, there are two phases to tunnel configuration.

The following procedure lists the recommended order in which you should configure an IPsec VPN tunnel:

1. Configure Phase 1 of the IPsec tunnel:
 - a. Configure an IKE Phase 1 proposal. (See “Example: Configuring an IKE Phase 1 Proposal (CLI)” on page 369.)

- b. Configure an IKE policy that references the proposal. (See “Example: Configuring an IKE Policy (CLI)” on page 370.)
 - c. Configure an IKE gateway that references the policy. (See “Example: Configuring an IKE Gateway (CLI)” on page 370.)
2. Configure Phase 2 of the IPsec tunnel:
 - a. Configure a Phase 2 proposal. (See “Example: Configuring an IPsec Phase 2 Proposal (CLI)” on page 373.)
 - b. Configure a policy that references the proposal. (See “Example: Configuring AutoKey IKE (CLI)” on page 374.)
 - c. Configure an Autokey IKE that references the policy and the gateway. (See “Example: Configuring AutoKey IKE (CLI)” on page 374.)
3. Update your global VPN settings. (See “Example: Configuring Global SPI and VPN Monitoring Features (CLI)” on page 375.)

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- VPN Overview on page 355
- Understanding IKE and IPsec Packet Processing on page 361
- Understanding Phase 1 of IKE Tunnel Negotiation on page 367
- Understanding Phase 2 of IKE Tunnel Negotiation on page 371
- Hub-and-Spoke VPN Configuration Overview on page 376

Phase 1 Proposals for IPsec VPNs

- Understanding Phase 1 of IKE Tunnel Negotiation on page 367
- Example: Configuring an IKE Phase 1 Proposal (CLI) on page 369
- Example: Configuring an IKE Policy (CLI) on page 370
- Example: Configuring an IKE Gateway (CLI) on page 370

Understanding Phase 1 of IKE Tunnel Negotiation

Phase 1 of an AutoKey IKE tunnel negotiation consists of the exchange of proposals for how to authenticate and secure the channel. The participants exchange proposals for acceptable security services such as:

- Encryption algorithms (DES and 3DES) and authentication algorithms (MD5 and SHA-1). (For more information, see “IPsec Security Protocols” on page 358.)
- A Diffie-Hellman (DH) group. (For more information, see “Diffie-Hellman Exchange” on page 358.)
- Preshared Key or RSA/DSA certificates. (For more information, see “IPsec Key Management” on page 357.)

A successful Phase 1 negotiation concludes when both ends of the tunnel agree to accept at least one set of the Phase 1 security parameters proposed and then process them. Juniper Networks devices support up to four proposals for Phase 1 negotiations, allowing you to define how restrictive a range of security parameters for key negotiation you will accept.

The predefined Phase 1 proposals that Junos OS provides are as follows:

- Standard—pre-g2-aes128-sha and pre-g2-3des-sha
- Compatible—pre-g2-3des-sha, pre-g2-3des-md5, pre-g2-des-sha, and pre-g2-des-md5
- Basic—pre-g1-des-sha and pre-g1-des-md5

You can also define custom Phase 1 proposals.



NOTE: If you are using the dynamic VPN feature, note that you must create a custom Phase 1 proposal. Predefined Phase 1 proposals are not available at this time.

Phase 1 exchanges can take place in either main or aggressive mode. You can choose your mode during IKE policy configuration.

This topic includes the following sections:

- Main Mode on page 368
- Aggressive Mode on page 368

Main Mode

In main mode, the initiator and recipient send three two-way exchanges (six messages total) to accomplish the following services:

- First exchange (messages 1 and 2)—Propose and accept the encryption and authentication algorithms.
- Second exchange (messages 3 and 4)—Execute a DH exchange, and the initiator and recipient each provide a pseudorandom number.
- Third exchange (messages 5 and 6)—Send and verify their identities.

The information transmitted in the third exchange of messages is protected by the encryption algorithm established in the first two exchanges. Thus, the participants' identities are not transmitted in the clear.

Aggressive Mode

In aggressive mode, the initiator and recipient accomplish the same objectives, but in only two exchanges, with a total of three messages:

- First message—The initiator proposes the SA, initiates a DH exchange, and sends a pseudorandom number and its IKE identity.

- Second message—The recipient accepts the SA; authenticates the initiator; and sends a pseudorandom number, its IKE identity, and, if using certificates, the recipient's certificate.
- Third message—The initiator authenticates the recipient, confirms the exchange, and, if using certificates, sends the initiator's certificate.

Because the participants' identities are exchanged in the clear (in the first two messages), aggressive mode does not provide identity protection.



NOTE: When a dialup VPN user negotiates an AutoKey IKE tunnel with a preshared key, aggressive mode must be used. Therefore, you must always use aggressive mode with the dynamic VPN feature. Note also that a dialup VPN user can use an e-mail address, a fully qualified domain name (FQDN), or an IP address as its IKE ID. A dynamic peer can use either an e-mail address or FQDN, but not an IP address.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- VPN Overview on page 355
- Example: Configuring an IKE Phase 1 Proposal (CLI) on page 369
- Example: Configuring an IKE Policy (CLI) on page 370
- Example: Configuring an IKE Gateway (CLI) on page 370

Example: Configuring an IKE Phase 1 Proposal (CLI)

When configuring Phase 1 of an IPsec tunnel using IKE, you first configure proposals, then policies, and finally the gateway. The following example-based instructions show how to create the proposal portion of the IKE gateway.

In Phase 1 proposal configuration, you must set the authentication method and authentication and encryption algorithms that will be used to open a secure channel between participants. In this example, you create an IKE proposal called `ike_prop_1` and specify that peers use preshared keys for encryption and decryption, and that they use Diffie-Hellman (DH) group 2 to produce the shared secret for the keys. You specify `md5` as the authentication algorithm and 3DES cypher block chaining (CBC) for encryption. And you specify that after 300 seconds the participants renegotiate a new security association (SA).



NOTE: When configuring a Phase 1 proposal for the dynamic VPN feature, note that you must set the authentication method to preshared keys.

To configure a Phase 1 proposal using the CLI editor:

```
user@host# set security ike proposal ike_prop_1 description "new ike proposal"
user@host# set security ike proposal ike_prop_1 authentication-method pre-shared-keys
user@host# set security ike proposal ike_prop_1 dh-group group2
user@host# set security ike proposal ike_prop_1 authentication-algorithm md5
```

```
user@host# set security ike proposal ike_prop_1 encryption-algorithm 3des-cbc
user@host# set security ike proposal ike_prop_1 lifetime-seconds 300
```

Use the following command to display information about IKE proposals:

```
user@host# show security ike
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Phase 1 of IKE Tunnel Negotiation on page 367
 - IPsec VPN Configuration Overview on page 366

Example: Configuring an IKE Policy (CLI)

When configuring Phase 1 of an IPsec tunnel using IKE, you first configure proposals, then policies, and finally the gateway. The following example-based instructions show how to create the policy portion of the IKE gateway.

During policy configuration, you must set the mode in which the Phase 1 channel will be negotiated, specify the type of key exchange to be used, and reference the Phase 1 proposal. In this example, you create a policy called `ike_pol_1`, specify that participants exchange proposals in aggressive mode, and reference the proposal called `ike_prop_1`. You specify that the preshared key be of type ASCII, and enter the key.



NOTE: When configuring an IKE policy for the dynamic VPN feature, note that you must set the mode to aggressive. Also note that you must use preshared keys rather than manual keys or certificates.

To configure an IKE policy using the CLI Editor:

```
user@host# set security ike policy ike_pol_1 mode aggressive
user@host# set security ike policy ike_pol_1 description "new ike policy"
user@host# set security ike policy ike_pol_1 proposals ike_prop_1
user@host# set security ike policy ike_pol_1 pre-shared-key ascii-text
"$9$UQiqf36A1RSTzRSreXxDik.Tzn/CuBI"
```

Use the following command to display information about this IKE policy:

```
user@host# show security ike policy ike_pol_1
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Phase 1 of IKE Tunnel Negotiation on page 367
 - IPsec VPN Configuration Overview on page 366

Example: Configuring an IKE Gateway (CLI)

When configuring Phase 1 of an IPsec tunnel using IKE, you first configure proposals, then policies, and finally the gateway. The following example-based instructions show how to create the IKE gateway.

When creating the gateway, you must reference the Phase 1 policy. In this example, you create an IKE gateway called `ike_gateway_1`, reference the policy `ike_pol_1`, and configure an IP address for the gateway. You configure dead peer detection (DPD) to send a DPD request packet when the device has not received traffic from a peer for 10 seconds, and to consider the peer unavailable after five sequences of waiting 10 seconds and sending a DPD request packet. You also specify `ge-0/0/0` as the outgoing interface.

To configure an IKE gateway using the CLI editor:

```
user@host# set security ike gateway ike_gateway_1 ike-policy ike_pol_1
user@host# set security ike gateway ike_gateway_1 address 1.1.1.2
user@host# set security ike gateway ike_gateway_1 dead-peer-detection interval 10
user@host# set security ike gateway ike_gateway_1 dead-peer-detection threshold 5
user@host# set security ike gateway ike_gateway_1 external-interface ge-0/0/0
```

Use the following command to display information about this IKE gateway:

```
user@host# show security ike gateway ike_gateway_1
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Phase 1 of IKE Tunnel Negotiation on page 367
 - IPsec VPN Configuration Overview on page 366

Phase 2 Proposals for IPsec VPNs

- Understanding Phase 2 of IKE Tunnel Negotiation on page 371
- Example: Configuring an IPsec Phase 2 Proposal (CLI) on page 373
- Example: Configuring an IPsec Policy (CLI) on page 373
- Example: Configuring AutoKey IKE (CLI) on page 374

Understanding Phase 2 of IKE Tunnel Negotiation

After the participants have established a secure and authenticated channel, they proceed through Phase 2, in which they negotiate the SAs to secure the data to be transmitted through the IPsec tunnel.

Similar to the process for Phase 1, the participants exchange proposals to determine which security parameters to employ in the SA. A Phase 2 proposal also includes a security protocol—either Encapsulating Security Payload (ESP) or Authentication Header (AH)—and selected encryption and authentication algorithms. The proposal can also specify a Diffie-Hellman (DH) group, if Perfect Forward Secrecy (PFS) is desired.

Regardless of the mode used in Phase 1, Phase 2 always operates in quick mode and involves the exchange of three messages.

Juniper Networks devices support up to four proposals for Phase 2 negotiations, allowing you to define how restrictive a range of tunnel parameters you will accept. The predefined Phase 2 proposals that Junos OS provides are as follows:

- Standard—`g2-esp-3des-sha` and `g2-esp-aes128-sha`

- Compatible—nopfs-esp-3des-sha, nopfs-esp-3des-md5, nopfs-esp-des-sha, and nopfs-esp-des-md5
- Basic—nopfs-esp-des-sha and nopfs-esp-des-md5

You can also define custom Phase 2 proposals.



NOTE: If you are using the dynamic VPN feature, note that you must create a custom Phase 2 proposal. Predefined Phase 2 proposals are not available at this time.

This topic includes the following sections:

- Proxy IDs on page 372
- Perfect Forward Secrecy on page 372
- Replay Protection on page 372

Proxy IDs

In Phase 2, the peers exchange proxy IDs. A proxy ID is a three-part tuple consisting of local IP address-remote IP address-service. The proxy ID for both peers must match, which means that the service specified in the proxy ID for both peers must be the same, and the local IP address specified for one peer must be the same as the remote IP address specified for the other peer.

Perfect Forward Secrecy

Perfect Forward Secrecy (PFS) is a method for deriving Phase 2 keys independent from and unrelated to the preceding keys. Alternatively, the Phase 1 proposal creates the key (the SKEYID_d key) from which all Phase 2 keys are derived. The SKEYID_d key can generate Phase 2 keys with a minimum of CPU processing. Unfortunately, if an unauthorized party gains access to the SKEYID_d key, all your encryption keys are compromised.

PFS addresses this security risk by forcing a new Diffie-Hellman key exchange to occur for each Phase 2 tunnel. Using PFS is thus more secure, although the rekeying procedure in Phase 2 might take slightly longer with PFS enabled.

Replay Protection

A replay attack occurs when somebody intercepts a series of packets and uses them later either to flood the system, causing a denial of service (DoS), or to gain entry to the trusted network. Junos OS provides a replay protection feature that enables devices to check every IPsec packet to see if it has been received previously. If packets arrive outside a specified sequence range, Junos OS rejects them. Use of this feature does not require negotiation, because packets are always sent with sequence numbers. You simply have the option of checking or not checking the sequence numbers.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - VPN Overview on page 355
 - Example: Configuring an IPsec Phase 2 Proposal (CLI) on page 373

- Example: Configuring an IPsec Policy (CLI) on page 373
- Example: Configuring AutoKey IKE (CLI) on page 374

Example: Configuring an IPsec Phase 2 Proposal (CLI)

When configuring Phase 2 of an IPsec tunnel, you first configure proposals, then policies, and finally the AutoKey IKE. The following example-based instructions show how to create the initial proposal.

In Phase 2 proposal configuration, you must create a proposal, specify a security protocol, and select authentication and encryption algorithms for the traffic that will flow through the tunnel. In this example, you create a proposal called `ipsec_prop_1`, specify ESP as the security protocol, and set `hmac-md5-96` as the authentication algorithm and `3des-cbc` as the encryption algorithm. You also specify that the security association (SA) terminates after 1,800 KB of data pass through it.

To configure an IPsec Phase 2 proposal using the CLI editor:

```
user@host# set security ipsec proposal ipsec_prop_1 description "new ipsec proposal"
user@host# set security ipsec proposal ipsec_prop_1 protocol esp
user@host# set security ipsec proposal ipsec_prop_1 authentication-algorithm
    hmac-md5-96
user@host# set security ipsec proposal ipsec_prop_1 encryption-algorithm 3des-cbc
user@host# set security ipsec proposal ipsec_prop_1 lifetime-seconds 1800
```

Use the following command to display information about this IKE proposal:

```
user@host# show security ipsec proposal ipsec_prop_1
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Phase 2 of IKE Tunnel Negotiation on page 371
 - IPsec VPN Configuration Overview on page 366

Example: Configuring an IPsec Policy (CLI)

When configuring Phase 2 of an IPsec tunnel, you first configure proposals, then policies, and finally the AutoKey IKE. The following example-based instructions show how to create the policy.

In Phase 2 IPsec policy configuration, you must create a policy and reference a Phase 2 proposal. In this example, you create a policy called `ipsec_pol_1` and reference the proposal `ipsec_prop_1`. You also configure Perfect Forward Secrecy (PFS) to use Diffie-Hellman (DH) group 2 as the method the device uses to generate the encryption key.

To configure an IPsec policy using the CLI editor:

```
user@host# set security ipsec policy ipsec_pol_1 description "new ipsec policy"
user@host# set security ipsec policy ipsec_pol_1 perfect-forward-secrecy keys group2
user@host# set security ipsec policy ipsec_pol_1 proposals ipsec_prop_1
```

Use the following command to display information about this IKE proposal:

```
user@host# show security ipsec policy ipsec_pol_1
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Phase 2 of IKE Tunnel Negotiation on page 371
 - IPsec VPN Configuration Overview on page 366

Example: Configuring AutoKey IKE (CLI)

When configuring Phase 2 of an IPsec tunnel, you first configure proposals, then policies, and finally the AutoKey IKE. The following example-based instructions show how to configure the AutoKey IKE.

In Phase 2 AutoKey IKE configuration, you must create a VPN tunnel name, specify a gateway, and reference a Phase 2 policy. If you are using route mode, you must bind the tunnel to an interface. In this example, you create a VPN tunnel named `vpn_1` and bind it to interface `st0.0`, and you specify `ike_gateway_1` as the gateway for the VPN tunnel and reference the IPsec policy `ipsec_pol_1`.

To configure an AutoKey IKE using the CLI editor:

```
user@host# set security ipsec vpn vpn_1 bind-interface st0.0
user@host# set security ipsec vpn vpn_1 ike gateway ike_gateway_1
user@host# set security ipsec vpn vpn_1 ike ipsec-policy ipsec_pol_1
```

Use the following command to display information about this IKE proposal:

```
user@host# show security ipsec vpn vpn_1
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Phase 2 of IKE Tunnel Negotiation on page 371
 - IPsec VPN Configuration Overview on page 366

Global SPI and VPN Monitoring Features

- Understanding Global SPI and VPN Monitoring Features on page 374
- Example: Configuring Global SPI and VPN Monitoring Features (CLI) on page 375

Understanding Global SPI and VPN Monitoring Features

You can monitor and maintain the efficient operation of your VPN using the following global VPN features:

- SPI—Peers in a security association (SA) can become unsynchronized when one of the peers fails. For example, if one of the peers reboots, it might send an incorrect security parameter index (SPI). You can enable the device to detect such an event and resynchronize the peers by configuring the bad SPI response feature.

- VPN monitoring—You can use the global VPN monitoring feature to periodically send Internet Control Message Protocol (ICMP) requests to the peer to determine if the peer is reachable.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - VPN Overview on page 355
 - IPsec VPN Configuration Overview on page 366
 - Example: Configuring Global SPI and VPN Monitoring Features (CLI) on page 375

Example: Configuring Global SPI and VPN Monitoring Features (CLI)

In this example, you configure the device to detect and respond five times to a bad IPsec security parameter index (SPI) before deleting the SA and initiating a new one. You also configure the device to monitor the VPN by sending Internet Control Message Protocol (ICMP) requests to the peer every 15 seconds, and to declare the peer unreachable after 15 unsuccessful pings.

To configure global VPN settings in the CLI editor:

```
user@host# set security ike respond-bad-spi 5
user@host# set security ipsec vpn-monitor-options interval 15 threshold 15
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Global SPI and VPN Monitoring Features on page 374
 - IPsec VPN Configuration Overview on page 366

Hub-and-Spoke VPNs

- Understanding Hub-and-Spoke VPNs on page 375
- Hub-and-Spoke VPN Configuration Overview on page 376
- Example: Configuring the Hub in a Hub-and-Spoke VPN (CLI) on page 377
- Example: Configuring Spoke 1 in a Hub-and-Spoke VPN (CLI) on page 380
- Example: Configuring Spoke 2 in a Hub-and-Spoke VPN (CLI) on page 381

Understanding Hub-and-Spoke VPNs

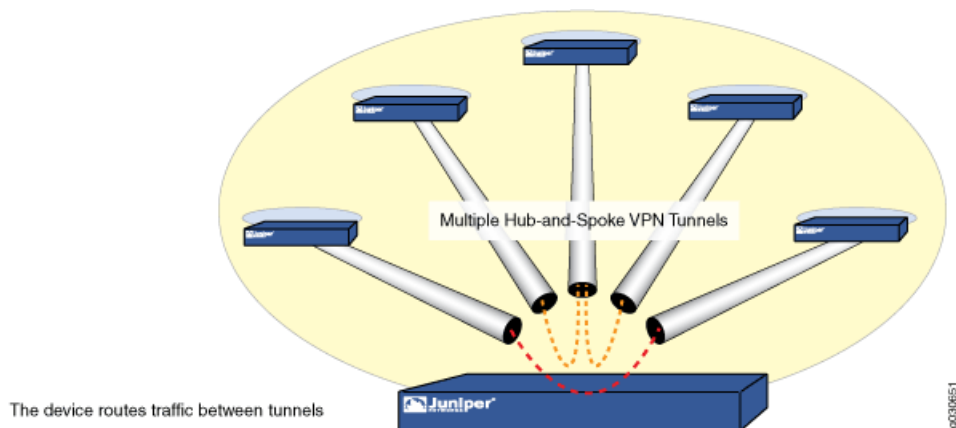
If you create two VPN tunnels that terminate at a device, you can set up a pair of routes so that the device directs traffic exiting one tunnel to the other tunnel. If both tunnels are contained within a single zone, you do not need to create a policy to permit the traffic to pass from one tunnel to the other. You only need to define the routes. Such an arrangement is known as *hub-and-spoke VPN*. (See Figure 49 on page 376.)

You can also configure multiple VPNs and route traffic between any two tunnels.



NOTE: SRX Series devices support only the route-based hub-and-spoke feature.

Figure 49: Multiple Tunnels in a Hub-and-Spoke VPN Configuration



- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Hub-and-Spoke VPN Configuration Overview on page 376
 - Example: Configuring the Hub in a Hub-and-Spoke VPN (CLI) on page 377
 - Example: Configuring Spoke 1 in a Hub-and-Spoke VPN (CLI) on page 380
 - Example: Configuring Spoke 2 in a Hub-and-Spoke VPN (CLI) on page 381

Hub-and-Spoke VPN Configuration Overview

For the hub router to be able to distinguish between packets going to and coming from the spoke routers, you must configure it with two routing instances.

The following instructions describe how to configure both the hub and the spokes in a hub-and-spoke VPN:

1. Configure Phase 1 of the IPsec tunnel:
 - a. Configure proposals. In Phase 1 proposal configuration, set the authentication method and authentication and encryption algorithms that will be used to open a secure channel between participants.



NOTE: When configuring a Phase 1 proposal for the dynamic VPN feature, note that you must set the authentication method to preshared keys.

- b. Configure policies. During policy configuration, you must set the mode in which the Phase 1 channel will be negotiated, specify the type of key exchange to be used, and reference the Phase 1 proposal.

- c. Configure the gateway. When creating the gateway, you must reference the Phase 1 policies.
2. Configure Phase 2 of the IPsec tunnel:
 - a. Configure proposals. In Phase 2 proposal configuration, you must create proposals for the two spokes, specify a security protocol, and select authentication and encryption algorithms for the traffic that will flow through the tunnel.
 - b. Configure policies. In Phase 2 IPsec policy configuration, you must create policies and reference the Phase 2 proposals.
 - c. Configure the AutoKey IKE. In Phase 2 AutoKey IKE configuration, you must create a VPN tunnel name, specify a gateway, and reference a Phase 2 policy. For route mode, you must bind the tunnel to an interface.
3. Configure a security policy.
4. Configure routing options.
5. Enable Next Hop Tunnel Binding (nhtb).

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Hub-and-Spoke VPNs on page 375
 - IPsec VPN Configuration Overview on page 366
 - Example: Configuring the Hub in a Hub-and-Spoke VPN (CLI) on page 377
 - Example: Configuring Spoke 1 in a Hub-and-Spoke VPN (CLI) on page 380
 - Example: Configuring Spoke 2 in a Hub-and-Spoke VPN (CLI) on page 381

Example: Configuring the Hub in a Hub-and-Spoke VPN (CLI)

The following example describes how to configure a hub in a hub-and-spoke VPN. The hub has two spokes (First and Third) and the configuration is for route-based VPNs.

In this example, you configure a Phase 1 tunnel on the hub using the following settings:

- You create two proposals called `first_ikeprop` and `third_ikeprop` and specify that peers use preshared keys for encryption and decryption and that they use Diffie-Hellman (DH) group 2 to produce the shared secret for the keys. You specify `md5` as the authentication algorithm and 3DES cypher block chaining (CBC) for encryption.
- You create two policies called `first_ikepol` and `third_ikepol`, specify that participants exchange proposals in aggressive mode, and reference the proposals called `first_ikeprop` and `third_ikeprop`. You specify that the preshared key be of type ASCII, and enter the key.
- You create IKE gateways called `ike_gateway_first` and `ike_gateway_third`, reference the policies `first_ikepol` and `third_ikepol`, and configure an IP address for the gateway.

Then, you configure a Phase 2 tunnel on the hub using the following settings:

- You create proposals called `first_ipsecprop` and `third_ipsecprop`, set `hmac-md5-96` as the authentication algorithm, and set `3des-cbc` as the encryption algorithm.
- You create two policies called `first_ipsecpol` and `third_ipsecpol` and reference the proposals `first_ipsecprop` and `third_ipsecprop`.
- You create VPN tunnels named `first_vpn` and `third_vpn` and bind them to interface `st0.0`, and you specify `ike_gateway_first` and `ike_gateway_third` as the gateways for the VPN tunnel and reference the IPsec policies `first_ipsecpol` and `third_ipsecpol`.

Finally, you configure a security policy and routing options and enable Next Hop Tunnel Binding (`nhtb`).

To configure the hub in a hub-and-spoke VPN:

1. Configure Phase 1 of the IPsec tunnel:

a. Configure IKE Phase 1 proposals:

```
user@host# set security ike proposal first_ikeprop authentication-method
pre-shared-keys
user@host# set security ike proposal first_ikeprop dh-group group2
user@host# set security ike proposal first_ikeprop authentication-algorithm
md5
user@host# set security ike proposal first_ikeprop encryption-algorithm 3des-cbc

user@host# set security ike proposal third_ikeprop authentication-method
pre-shared-keys
user@host# set security ike proposal third_ikeprop dh-group group2
user@host# set security ike proposal third_ikeprop authentication-algorithm
md5
user@host# set security ike proposal third_ikeprop encryption-algorithm 3des-cbc
```

b. Configure IKE policies (and reference the proposals):

```
user@host# set security ike policy first_ikepol mode main
user@host# set security ike policy first_ikepol proposals first_ikeprop
user@host# set security ike policy first_ikepol pre-shared-key ascii-text
"$9$xFU-b2ZUH5Qn4aQn/CB17-V"

user@host# set security ike policy third_ikepol mode main
user@host# set security ike policy third_ikepol proposals third_ikeprop
user@host# set security ike policy third_ikepol pre-shared-key ascii-text
"$9$GvjKPFnCB1c5Q1cyLXUjH"
```

c. Configure the IKE gateway (and reference the policy):

```
user@host# set security ike gateway first ike-policy first_ikepol
user@host# set security ike gateway first address 4.4.4.2
user@host# set security ike gateway first external-interface ge-0/0/0.0

user@host# set security ike gateway third ike-policy third_ikepol
user@host# set security ike gateway third address 2.2.2.1
```

```
user@host# set security ike gateway third external-interface ge-0/0/3.0
```

2. Configure Phase 2 of the IPsec tunnel:

a. Configure Phase 2 proposals:

```
user@host# set security ipsec proposal first_ipsecprop protocol esp
user@host# set security ipsec proposal first_ipsecprop authentication-algorithm
    hmac-md5-96
user@host# set security ipsec proposal first_ipsecprop encryption-algorithm
    3des-cbc
```

```
user@host# set security ipsec proposal third_ipsecprop protocol esp
user@host# set security ipsec proposal third_ipsecprop authentication-algorithm
    hmac-md5-96
user@host# set security ipsec proposal third_ipsecprop encryption-algorithm
    3des-cbc
```

b. Configure policies (and reference proposals):

```
user@host# set security ipsec policy first_ipsecpol perfect-forward-secrecy keys
    group1
user@host# set security ipsec policy first_ipsecpol proposals first_ipsecprop

user@host# set security ipsec policy third_ipsecpol perfect-forward-secrecy
    keys group1
user@host# set security ipsec policy third_ipsecpol proposals third_ipsecprop
```

c. Configure AutoKey IKE (and reference the policy and gateway):

```
user@host# set security ipsec vpn first_vpn bind-interface st0.0
user@host# set security ipsec vpn first_vpn ike gateway first
user@host# set security ipsec vpn first_vpn ike ipsec-policy first_ipsecpol
user@host# set security ipsec vpn first_vpn establish-tunnels immediately

user@host# set security ipsec vpn third_vpn bind-interface st0.0
user@host# set security ipsec vpn third_vpn ike gateway third
user@host# set security ipsec vpn third_vpn ike ipsec-policy third_ipsecpol
user@host# set security ipsec vpn third_vpn establish-tunnels immediately
```

3. Configure the security policy:

```
user@host# set security policies default-policy permit-all
```

4. Configure routing options:

```
user@host# set routing-options static route 1.1.1.0/24 next-hop st0.0
user@host# set routing-options static route 3.1.1.0/24 next-hop st0.0
```

5. Enable Next Hop Tunnel Binding (nhtb):

```
user@host# set interfaces st0 unit 0 multipoint
```

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Hub-and-Spoke VPNs on page 375
- Hub-and-Spoke VPN Configuration Overview on page 376
- Example: Configuring Spoke 1 in a Hub-and-Spoke VPN (CLI) on page 380
- Example: Configuring Spoke 2 in a Hub-and-Spoke VPN (CLI) on page 381

Example: Configuring Spoke 1 in a Hub-and-Spoke VPN (CLI)

The following example describes how to configure spoke 1 in a hub-and-spoke VPN. The hub has two spokes (First and Third) and the configuration is for route-based VPNs. Follow the same process to configure spoke First as you did to configure the hub:

1. Configure Phase 1 of the IPsec tunnel:
 - a. Configure IKE Phase 1 proposals:

```
user@host# set security ike proposal ike_prop authentication-method pre-shared-keys
user@host# set security ike proposal ike_prop dh-group group2
user@host# set security ike proposal ike_prop authentication-algorithm md5
user@host# set security ike proposal ike_prop encryption-algorithm 3des-cbc
```
 - b. Configure IKE policies (and reference the proposals):

```
user@host# set security ike policy ike_pol mode main
user@host# set security ike policy ike_pol proposals ike_prop
user@host# set security ike policy ike_pol pre-shared-key ascii-text "$9$va38xd24Zk.5bs.5QFAtm8X"
```
 - c. Configure IKE gateway (and reference the policy):

```
user@host# set security ike gateway first ike-policy ike_pol
user@host# set security ike gateway first address 4.4.4.1
user@host# set security ike gateway first external-interface fe-2/0/0.0
```
2. Configure Phase 2 of the IPsec tunnel:
 - a. Configure Phase 2 proposals:

```
user@host# set security ipsec proposal ipsec_prop protocol esp
user@host# set security ipsec proposal ipsec_prop authentication-algorithm hmac-md5-96
user@host# set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
```
 - b. Configure policies (and reference proposals):

```
user@host# set security ipsec policy ipsec_pol perfect-forward-secrecy keys group1
user@host# set security ipsec policy ipsec_pol proposals ipsec_prop
```
 - c. Configure AutoKey IKE (and reference the policy and gateway):

```
user@host# set security ipsec vpn first_vpn bind-interface st0.0
user@host# set security ipsec vpn first_vpn ike gateway gate
user@host# set security ipsec vpn first_vpn ike ipsec-policy ipsec_pol
```
3. Configure the security policy:

```
user@host# set security policies default-policy permit-all
```
4. Configure routing options:

```
user@host# set routing-options static route 1.1.1.0/24 next-hop 7.7.7.1
```

Related Topics • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Understanding Hub-and-Spoke VPNs on page 375
- Hub-and-Spoke VPN Configuration Overview on page 376
- Example: Configuring the Hub in a Hub-and-Spoke VPN (CLI) on page 377
- Example: Configuring Spoke 2 in a Hub-and-Spoke VPN (CLI) on page 381

Example: Configuring Spoke 2 in a Hub-and-Spoke VPN (CLI)

The following example describes how to configure spoke 2 in a hub-and-spoke VPN. The hub has two spokes (First and Third) and the configuration is for route-based VPNs. Follow the same process to configure spoke Third as you did to configure spoke First:

1. Configure Phase 1 of the IPsec tunnel:
 - a. Configure IKE Phase 1 proposals:


```
user@host# set security ike proposal ike_prop authentication-method
pre-shared-keys
user@host# set security ike proposal ike_prop dh-group group2
user@host# set security ike proposal ike_prop authentication-algorithm md5
user@host# set security ike proposal ike_prop encryption-algorithm 3des-cbc
```
 - b. Configure IKE policies (and reference the proposals):


```
user@host# set security ike policy ike_pol mode main
user@host# set security ike policy ike_pol proposals ike_prop
user@host# set security ike policy ike_pol pre-shared-key ascii-text
"$9$JrUi.QF/OBEP5BEcyW8ZUj"
user@host# set security ike gateway gate ike-policy ike_pol
```
 - c. Configure IKE gateway (and reference the policy):


```
user@host# set security ike gateway third address 2.2.2.2
user@host# set security ike gateway third external-interface ge-0/0/3.0
```
2. Configure Phase 2 of the IPsec tunnel:
 - a. Configure Phase 2 proposals:


```
user@host# set security ipsec proposal ipsec_prop protocol esp
user@host# set security ipsec proposal ipsec_prop authentication-algorithm
hmac-md5-96
user@host# set security ipsec proposal ipsec_prop encryption-algorithm 3des-cbc
```
 - b. Configure policies (and reference proposals):


```
user@host# set security ipsec policy ipsec_pol perfect-forward-secrecy keys
group1
user@host# set security ipsec policy ipsec_pol proposals ipsec_prop
```
 - c. Configure AutoKey IKE (and reference the policy and gateway):


```
user@host# set security ipsec vpn first_vpn bind-interface st0.0
user@host# set security ipsec vpn first_vpn ike gateway gate
user@host# set security ipsec vpn first_vpn ike ipsec-policy ipsec_pol
```
3. Configure the security policy:

```
user@host# set security policies default-policy permit-all
```

4. Configure routing options:

```
user@host# set routing-options static route 3.1.1.0/24 next-hop 7.7.7.1
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Hub-and-Spoke VPNs on page 375
 - Hub-and-Spoke VPN Configuration Overview on page 376
 - Example: Configuring the Hub in a Hub-and-Spoke VPN (CLI) on page 377
 - Example: Configuring Spoke 1 in a Hub-and-Spoke VPN (CLI) on page 380

CHAPTER 19

Public Key Cryptography for Certificates

- Understanding Public Key Infrastructure on page 383
- Certificates and Certificate Authority on page 385
- Self-Signed Certificates on page 398
- Certificate Revocation Lists on page 400

Understanding Public Key Infrastructure

Public key infrastructure (PKI) refers to the hierarchical structure of trust required for the successful implementation of public key cryptography. To verify the trustworthiness of a certificate, you must be able to track a path of certified certificate authorities (CAs) from the one issuing your local certificate to the root authority of a CA domain. See Figure 50 on page 384.

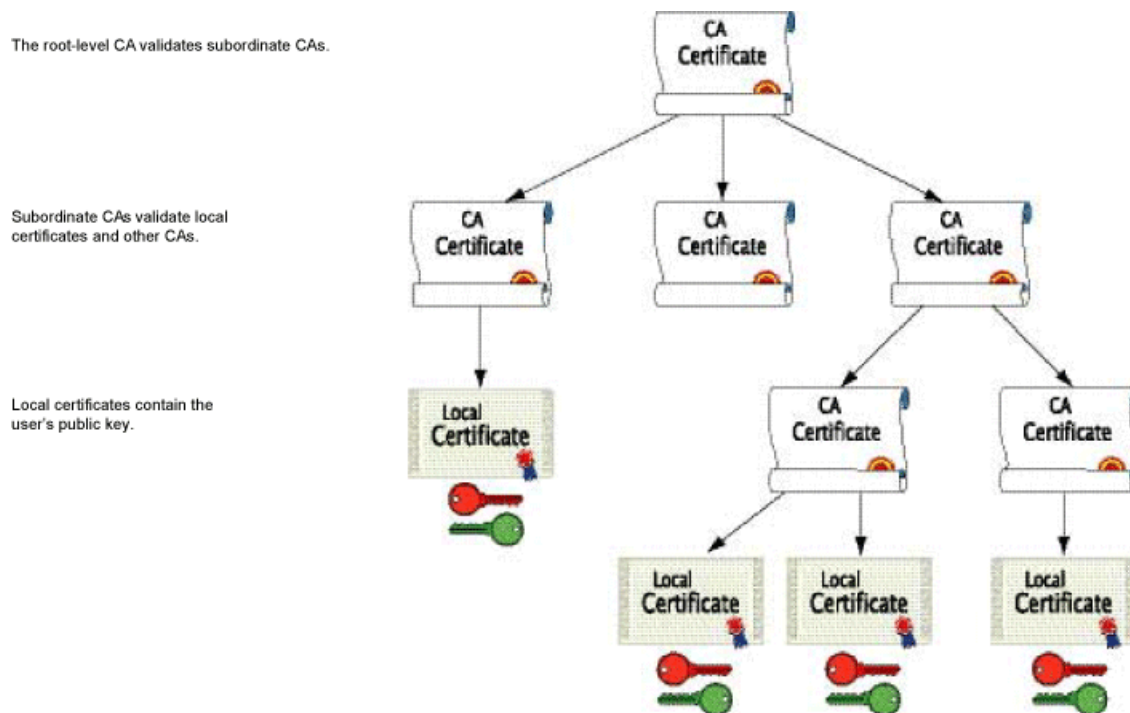
This topic includes the following sections:

- PKI Hierarchy for a Single CA Domain or Across Domains on page 383
- PKI Management and Implementation on page 385

PKI Hierarchy for a Single CA Domain or Across Domains

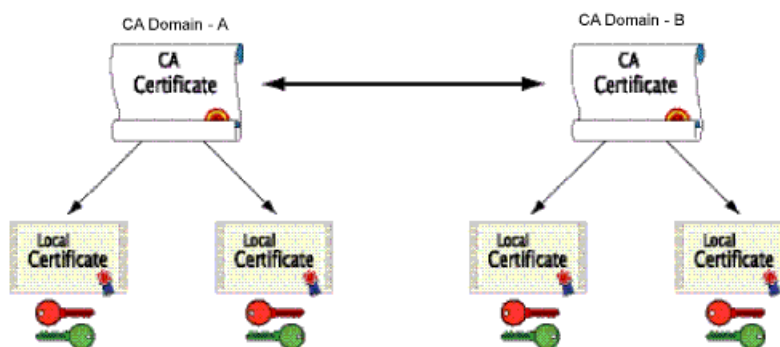
Figure 50 on page 384 shows the structure of a single-domain certificate authority.

Figure 50: PKI Hierarchy of Trust—CA Domain



If certificates are used solely within an organization, that organization can have its own CA domain within which a company CA issues and validates certificates for its employees. If that organization later wants its employees to exchange their certificates with certificates from another CA domain (for example, with employees at another organization that has its own CA domain), the two CAs can develop cross-certification by agreeing to trust the authority of each other. In this case, the PKI structure does not extend vertically but does extend horizontally. See Figure 51 on page 384.

Figure 51: Cross-Certification



Users in the CA domain A can use their certificates and key pairs with users in CA domain B because the CA's have cross-certified each other.

PKI Management and Implementation

For convenience and practicality, PKI must be transparently managed and implemented. Toward this goal, Junos OS supports the following features:

- Generates a public-private key pair.
- Loads multiple local certificates from different CAs.
- Delivers a certificate when establishing an IPsec tunnel.
- Validates a certificate path upward through eight levels of CA authorities in the PKI hierarchy.
- Supports the Public-Key Cryptography Standards #7 (PKCS #7) cryptographic . As a result, the device can accept X.509 certificates and certificate revocation lists (CRLs) packaged within a PKCS #7 envelope.



NOTE: Junos OS supports a PKCS #7 file size of up to 7 KB.

- Retrieves CRLs online retrieval through Lightweight Directory Access Protocol (LDAP) or Hypertext Transfer Protocol (HTTP).

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Public Key Cryptography on page 389
 - Understanding Certificates on page 386
 - Understanding Certificate Revocation Lists on page 401
 - Understanding Self-Signed Certificates on page 398

Certificates and Certificate Authority

- Understanding Certificates on page 386
- Digital Certificates Configuration Overview on page 387
- Public-Private Key Pairs on page 389
- Certificate Authority Profiles on page 390
- Certificate Enrollment on page 391
- Example: Generating a Local Certificate Request Manually (CLI) on page 394
- Example: Loading CA and Local Certificates Manually (CLI) on page 395
- Example: Reenrolling Local Certificates Automatically (CLI) on page 396
- Deleting Certificates (CLI Procedure) on page 397

Understanding Certificates

A digital certificate is an electronic means for verifying your identity through a trusted third party, known as a certificate authority (CA). Alternatively, you can use a self-signed certificate to attest to your identity.

The CA server you use can be owned and operated by an independent CA or by your own organization, in which case you become your own CA. If you use an independent CA, you must contact them for the addresses of their CA and certificate revocation list (CRL) servers (for obtaining certificates and certificate revocation lists) and for the information they require when submitting personal certificate requests. When you are your own CA, you determine this information yourself.



NOTE: The following CAs are supported: Entrust, Microsoft, and Verisign.

This topic includes the following sections:

- Certificate Signatures on page 386
- Certificate Verification on page 386
- Internet Key Exchange on page 387

Certificate Signatures

The CA that issues a certificate uses a Message Digest 5 (MD5) or Secure Hash Algorithm 1 (SHA-1) to generate a digest, and then “signs” the certificate by encrypting the digest with its private key. The result is a digital signature. The CA then makes the digitally signed certificate available for download to the person who requested it. Figure 52 on page 387 illustrates this process.

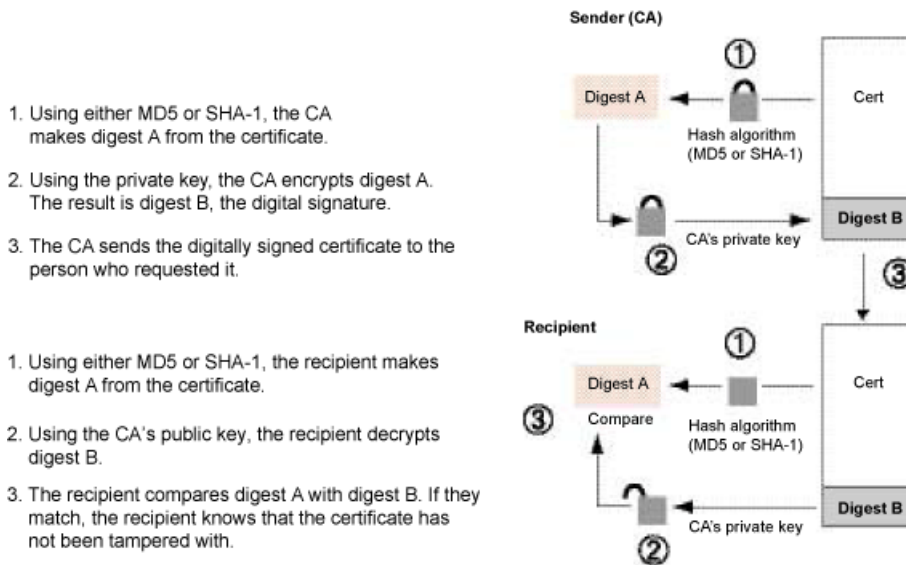
Certificate Verification

The recipient of the certificate generates another digest by applying the same MD5 or SHA-1 hash algorithm to the certificate file, then uses the CA's public key to decrypt the digital signature. By comparing the decrypted digest with the digest just generated, the recipient can confirm the integrity of the CA's signature and, by extension, the integrity of the accompanying certificate. Figure 52 on page 387 illustrates this process.



NOTE: If the issuer of the end-entity (EE) certificate is not a root certificate, up to eight levels are verified. Revocation status of each certificate in the verification chain is also verified. A certificate revocation status is considered “good” when its serial number is not in the CRL, which satisfies the refresh requirement per CA profile.

Figure 52: Digital Signature Verification



Internet Key Exchange

The procedure for digitally signing messages sent between two participants in an Internet Key Exchange (IKE) session is similar to digital certificate verification, with the following differences:

- Instead of making a digest from the CA certificate, the sender makes it from the data in the IP packet payload.
- Instead of using the CA's public-private key pair, the participants use the sender's public-private key pair.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Digital Certificates Configuration Overview on page 387
- Example: Generating a Public-Private Key Pair (CLI) on page 390
- Example: Generating a Local Certificate Request Manually (CLI) on page 394
- Example: Loading CA and Local Certificates Manually (CLI) on page 395

Digital Certificates Configuration Overview

Digital certificates authenticate your identity when establishing secure virtual private network (VPN) connections.

To use a digital certificate to authenticate your identity when establishing a secure VPN connection, you must first do the following:

- Obtain a certificate authority (CA) certificate from which you intend to obtain a personal certificate, and then load the CA certificate onto the device.

The CA certificate can contain a certificate revocation list (CRL) to identify invalid certificates.

- Obtain a local certificate (also known as a personal certificate) from the CA whose CA certificate you have previously loaded, and then load the local certificate in the device. The local, or end-entity (EE), certificate establishes the identity of the Juniper Networks device with each tunnel connection.

You can obtain CA and local certificates manually, or online using the Simple Certificate Enrollment Protocol (SCEP). Certificates are verifiable and renewable, and you can delete them when they are no longer needed.

This topic includes the following sections:

- [Enabling Digital Certificates Online: Configuration Overview](#) on page 388
- [Manually Generating Digital Certificates: Configuration Overview](#) on page 388
- [Verifying the Validity of a Certificate: Configuration Overview](#) on page 389
- [Deleting a Certificate: Configuration Overview](#) on page 389

Enabling Digital Certificates Online: Configuration Overview

SCEP uses the online method to request digital certificates. To obtain a certificate online:

1. Generate a key pair in the device. See “Example: Generating a Public-Private Key Pair (CLI)” on page 390.
2. Create a CA profile containing information specific to a CA. You can have multiple CA profiles on the device. For example, you might have one profile for Microsoft and one for Entrust. See “Example: Configuring a Certificate Authority Profile (CLI)” on page 391.
3. Enroll the CA certificate onto the device. See “Enrolling a CA Certificate Online (CLI Procedure)” on page 392.
4. Obtain a local certificate (also known as a personal certificate) online from the CA whose CA certificate you have previously loaded. See “Example: Enrolling a Local Certificate Online (CLI)” on page 392.
5. Configure automatic reenrollment. See “Example: Configuring SecurID User Authentication” on page 317.

Manually Generating Digital Certificates: Configuration Overview

To obtain digital certificates manually:

1. Generate a key pair in the device. See “Example: Generating a Public-Private Key Pair (CLI)” on page 390.
2. Create a CA profile containing information specific to a CA. You can have multiple CA profiles on the device. For example, you might have one profile for Microsoft and one for Entrust. See “Example: Configuring a Certificate Authority Profile (CLI)” on page 391.
3. Generate a certificate request using the key pair, and manually copy that request and paste it into the appropriate field at the CA website to obtain a personal

certificate (also known as a local certificate). See “Example: Generating a Local Certificate Request Manually (CLI)” on page 394.

4. Load the certificate onto the device. See “Example: Loading CA and Local Certificates Manually (CLI)” on page 395.
5. Configure automatic reenrollment. See “Example: Configuring SecurID User Authentication” on page 317.
6. If necessary, load the certificate's CRL on the device. See “Example: Manually Loading a CRL onto the Device (CLI)” on page 401.

Verifying the Validity of a Certificate: Configuration Overview

To verify the validity of a certificate manually, see “Example: Verifying Certificate Validity (CLI)” on page 402.

Deleting a Certificate: Configuration Overview

To delete a certificate or a certificate revocation list (CRL), see “Deleting Certificates (CLI Procedure)” on page 397 and “Deleting a Loaded CRL (CLI Procedure)” on page 403.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Certificates on page 386
 - Understanding Certificate Revocation Lists on page 401
 - Understanding Public Key Infrastructure on page 383
 - Understanding Self-Signed Certificates on page 398

Public-Private Key Pairs

- Understanding Public Key Cryptography on page 389
- Example: Generating a Public-Private Key Pair (CLI) on page 390

Understanding Public Key Cryptography

The public-private key pairs used in public key cryptography play an important role in the use of digital certificates. A public-private key pair encrypts and decrypts data. Data encrypted with a public key, which the owner makes available to the public, can be decrypted with the corresponding private key only, which the owner keeps secret and protected. For example, if Alice wants to send Bob an encrypted message, Alice can encrypt it with Bob's public key and send it to him. Bob then decrypts the message with his private key.

The reverse process is also useful: encrypting data with a private key and decrypting it with the corresponding public key. This process is known as creating a digital signature. For example, if Alice wants to present her identity as the sender of a message, she can encrypt the message with her private key and send the message to Bob. Bob then decrypts the message with Alice's public key, thus verifying that Alice is indeed the sender.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Certificates on page 386

- Example: Generating a Public-Private Key Pair (CLI) on page 390
- Digital Certificates Configuration Overview on page 387

Example: Generating a Public-Private Key Pair (CLI)

When you generate a public-private key pair, the device automatically saves the key pair in a file in the certificate store, where it is subsequently used in certificate request commands.

If the device renews a great number of certificates at once, thus using up keys rapidly, it might run out of pregenerated keys and have to generate them promptly for each new request. In this case, the generation of keys might affect the performance of the device, especially in a high-availability environment where the performance of the device might slow down for a number of minutes.

You must have root-level privileges to generate a public-private key pair. When you generate a public-private key pair on the device, the generated key pair is saved as **certificate-id.priv**.

To generate a public-private key pair named, for example, **ca-ipsec**, with a key size of 1024 bits, enter the following command:

```
user@host> request security pki generate-key-pair certificate-id ca-ipsec
```



NOTE: The default RSA key size is 1024 bits. If you are using the Simple Certificate Enrollment Protocol (SCEP), Junos OS supports RSA only.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Public Key Cryptography on page 389
- Example: Verifying Certificate Validity (CLI) on page 402

Certificate Authority Profiles

- Understanding Certificate Authority Profiles on page 390
- Example: Configuring a Certificate Authority Profile (CLI) on page 391

Understanding Certificate Authority Profiles

A certificate authority (CA) profile configuration contains information specific to a CA. You can have multiple CA profiles on the device. For example, you might have one profile for Microsoft and one for Entrust. Each profile is associated with a CA certificate. If you want to load a new CA certificate without removing the older one, you must create a new CA profile (for example, Microsoft-2008).

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Certificates on page 386
- Understanding Certificate Revocation Lists on page 401

- Digital Certificates Configuration Overview on page 387
- Example: Configuring a Certificate Authority Profile (CLI) on page 391

Example: Configuring a Certificate Authority Profile (CLI)

To configure a CA profile:

1. Create a CA profile. For example, the following command creates a CA profile called **ca-profile-ipsec** with CA identity **microsoft-2008**, specifies that the CRL be refreshed every 48 hours, and indicates the location to retrieve the CRL from is **http://www.my-ca.com**:

```
user@host# set security pki ca-profile ca-profile-ipsec ca-identity microsoft-2008
revocation-check crl refresh-interval 48 url http://www.my-ca.com
```

2. Specify the number of times a device resends a certificate request for online enrollment when attempts to enroll in Step 1 fail. For example, the following command sets the enrollment retry to 20 times:

```
user@host# set security pki ca-profile ca-profile-ipsec enrollment retry 20
```

The default value for **retry** is 10.

3. Specify the time interval in seconds between attempts to automatically enroll the CA certificate online. For example, the following command specifies automatic certificate polling every 30 minutes:

```
user@host# set security pki ca-profile ca-profile-ipsec enrollment retry-interval 1800
```

If you configure **retry** only without configuring a **retry interval**, then the default **retry interval** is 900 seconds (15 minutes). If you do not configure **retry** or a **retry interval**, then there is no polling.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Example: Generating a Public-Private Key Pair (CLI) on page 390
 - Enrolling a CA Certificate Online (CLI Procedure) on page 392
 - Example: Enrolling a Local Certificate Online (CLI) on page 392
 - Understanding Certificate Authority Profiles on page 390
 - Deleting Certificates (CLI Procedure) on page 397

Certificate Enrollment

- Understanding Online CA Certificate Enrollment on page 391
- Enrolling a CA Certificate Online (CLI Procedure) on page 392
- Example: Enrolling a Local Certificate Online (CLI) on page 392

Understanding Online CA Certificate Enrollment

With Simple Certificate Enrollment Protocol (SCEP), you can configure your Juniper Networks device to obtain a certificate authority (CA) certificate online and start the

online enrollment for the specified certificate ID. The CA public key verifies certificates from remote peers.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Public Key Cryptography on page 389
 - Understanding Certificates on page 386
 - Enrolling a CA Certificate Online (CLI Procedure) on page 392
 - Example: Enrolling a Local Certificate Online (CLI) on page 392

Enrolling a CA Certificate Online (CLI Procedure)

Before you begin:

1. Generate a public and private key pair. See “Example: Generating a Public-Private Key Pair (CLI)” on page 390.
2. Configure a CA profile. See “Example: Configuring a Certificate Authority Profile (CLI)” on page 391.

To enroll a CA certificate online:

1. Use the following command to get the CA certificate online using SCEP. The attributes required to reach the CA server are obtained from the defined CA profile.

```
user@host> request security pki ca-certificate enroll ca-profile ca-profile_name
```

The command is processed synchronously to provide the fingerprint of the received CA certificate.

Fingerprint:

```
e6:fa:d6:da:e8:8d:d3:00:e8:59:12:e1:2c:b9:3c:c0:9d:6c:8f:8d (sha1)
```

```
82:e2:dc:ea:48:4c:08:9a:fd:b5:24:b0:db:c3:ba:59 (md5)
```

```
Do you want to load the above CA certificate ? [yes,no]
```

2. You must confirm that the correct certificate is loaded. The CA certificate is loaded only when you type **yes** at the CLI prompt. For more information on the certificate, such as the bit length of the key pair, use the command **show security pki ca-certificate** described in the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Digital Certificates Configuration Overview on page 387
 - Understanding Online CA Certificate Enrollment on page 391
 - Example: Verifying Certificate Validity (CLI) on page 402

Example: Enrolling a Local Certificate Online (CLI)

With SCEP, you can configure your Juniper Networks device to obtain a local certificate online and start the online enrollment for the specified certificate ID.

Before you begin:

1. Generate a public and private key pair. See “Example: Generating a Public-Private Key Pair (CLI)” on page 390.
2. Configure a certificate authority (CA) profile. See “Example: Configuring a Certificate Authority Profile (CLI)” on page 391.
3. Enroll a CA certificate. See “Enrolling a CA Certificate Online (CLI Procedure)” on page 392

To configure the device for online enrollment:

1. Specify the CA profile—for example, **wincs-5**—and specify the CA location for your device to send the SCEP-based certificate enrollment requests. To specify the CA location by naming the CA URL, include the **url** statement. For example:


```
user@host# set security pki ca-profile winsc-5 enrollment url
http://10.155.8.1/certsrv/mscep/mscep.dll
```
2. Use the **request security pki local-certificate enroll** command to start the online enrollment for the specified certificate ID. You must specify the CA profile name (for example, **wincs-5**), the certificate ID (for example, **qqq**), and the following information:



NOTE: SCEP sends a PKCS #10 format certificate request enveloped in PKCS #7 format.

- Specify the **challenge CA password** for certificate enrollment and revocation—for example, **aaa**. If the CA does not provide the challenge password, then choose your own password.
- Specify at least one of the following values:
 - Enter the domain name to identify the certificate owner in Internet Key Exchange (IKE) negotiations—for example, **qqq.juniper.net**.
 - Specify the identity of the certificate owner for IKE negotiation with the e-mail statement—for example, **qqq@juniper.net**.
 - Enter an IP address if the device is configured for a static IP address—for example, **10.10.10.10**.
- Specify the subject name in the distinguished name format in quotation marks, including the domain component (DC), common name (CN), organizational unit name (OU), organization name (O), locality (L), state (ST), and country (C).

For example:

```
user@host> request security pki local-certificate enroll ca-profile winsc-5
certificate-id qqq challenge-password aaa domain-name qqq.juniper.net
email qqq@juniper.net ip-address 10.10.10.10 subject DC=juniper,
CN=router3, OU=marketing, O=juniper, L=sunnyvale, ST=california, C=us
```

The device certificate is obtained and the online enrollment begins for the certificate ID. The command is processed asynchronously.

The device certificate is obtained and the online enrollment begins for the certificate ID. The command is processed asynchronously.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Online CA Certificate Enrollment on page 391
 - Digital Certificates Configuration Overview on page 387
 - Example: Generating a Local Certificate Request Manually (CLI) on page 394
 - Example: Loading CA and Local Certificates Manually (CLI) on page 395
 - Example: Verifying Certificate Validity (CLI) on page 402

Example: Generating a Local Certificate Request Manually (CLI)

When you create a local certificate request, the device generates a CA certificate in PKCS #10 format from a key pair you previously generated using the same certificate ID.

Before you begin:

1. Generate a public and private key. See “Example: Generating a Public-Private Key Pair (CLI)” on page 390.
2. Create a CA profile. See “Understanding Certificate Authority Profiles” on page 390.

A subject name is associated with the local certificate request in the form of a common name (CN), organizational unit (OU), organization (O), locality (L), state (ST), country (C), and domain component (DC). Additionally, a subject alternative name is associated in the following form:

- IP address
- E-mail address
- Fully qualified domain name (FQDN)



NOTE: Some CAs do not support an e-mail address as the domain name in a certificate. If you do not include an e-mail address in the local certificate request, you cannot use an e-mail address as the local IKE ID when configuring the device as a dynamic peer. Instead, you can use a fully qualified domain name (if it is in the local certificate), or you can leave the local ID field empty. If you do not specify a local ID for a dynamic peer, enter the *hostname.domain-name* of that peer on the device at the other end of the IPsec tunnel in the peer ID field.

To generate a certificate request using the certificate ID (**ca-ipsec**) of a public-private key pair you previously generated and specifying the domain name **juniper.net** and the associated common name **abc**:

1. Enter the following command:

```
user@host> request security pki generate-certificate-request certificate-id ca-ipsec  
domain-name juniper.net subject CN=abc
```

The following certificate request is displayed in PEM format.

```
Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIHxMIGcAgEAMA4xDDAKBgNVBAMTA2htMTBcMA0GCSqGSIb3DQEBAQUAA0sAMEgC
QQCbhaiWzmctH0ZD1dCn+mSNM62kyiSgc4cmN68U/j9E109/DgGoMny2y+RYA1xU
sr4B0NedGrZZJx5L1sIYjHr/AgMBAAgKTAkBqkqhkiG9w0BCQ4xGjAYMBYGA1Ud
EQQPMA2CC2p1bm1wZXIubmVMA0GCSqGSIb3DQEBBQUAA0EA1eLR6Hp2ity8Dugs
MW4HI6SxfwMc2eYM5Nj2UhwpeEpsce77dUBZrIKdehAg1i7vwNshGIIuhHjEaFzf0
hpM3tA==
-----END CERTIFICATE REQUEST-----
Fingerprint:
9e:d5:7d:44:e8:e7:b6:d7:4b:58:d4:4e:2b:fb:c6:b2:4b:b7:8b:82 (sha1)
b0:8d:c7:6d:41:d5:58:61:dc:a0:3e:4e:d6:39:02:d7 (md5)
```

2. Copy the generated certificate request and paste it into the appropriate field at the CA website to obtain a local certificate. Refer to the CA server documentation to determine where to paste the certificate request.

When PKCS #10 content is displayed, the MD5 hash and SHA-1 hash of the PKCS #10 file is also displayed. For more information on the certificate, such as the bit length of the key pair, use the command **show security pki certificate-request** described in the *Junos OS CLI Reference*.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Digital Certificates Configuration Overview on page 387
- Example: Loading CA and Local Certificates Manually (CLI) on page 395
- Example: Reenrolling Local Certificates Automatically (CLI) on page 396
- Example: Verifying Certificate Validity (CLI) on page 402
- Example: Checking Certificate Validity Using CRLs (CLI) on page 403

Example: Loading CA and Local Certificates Manually (CLI)

After you download certificates from a CA, you transfer them to the device (for example, using FTP), and then load them.

Before you begin:

1. Generate a public-private key pair. See “Example: Generating a Public-Private Key Pair (CLI)” on page 390.
2. Create a CA profile. See “Understanding Certificate Authority Profiles” on page 390.
3. Generate a certificate request. See “Example: Generating a Local Certificate Request Manually (CLI)” on page 394.

You can load the following certificate files onto a device running Junos OS:

- A local or end-entity (EE) certificate that identifies your local device. This certificate is your public key.
- A CA certificate that contains the CA's public key.

- A CRL that lists any certificates revoked by the CA.



NOTE: You can load multiple EE certificates onto the device.

In this example, you have downloaded the following certificates and saved them to the `/var/tmp/` directory on the device:

- `local.cert`
- `ca.cert`

To load the certificate files onto a device:

1. To load the local certificate called `local.cert` from the `/var/tmp` directory on the device, enter the following command:

```
user@host> request security pki local-certificate load certificate-id local.cert filename  
/var/tmp/local.cert
```

2. To load the CA certificate called `ca.cert` from the `/var/tmp` directory on the device, enter the following command. The CA profile is called `ca-profile-ipsec`.

```
user@host> request security pki ca-certificate load ca-profile ca-profile-ipsec  
filename /var/tmp/ca.cert
```

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Digital Certificates Configuration Overview on page 387
- Example: Reenrolling Local Certificates Automatically (CLI) on page 396
- Example: Verifying Certificate Validity (CLI) on page 402
- Example: Checking Certificate Validity Using CRLs (CLI) on page 403

Example: Reenrolling Local Certificates Automatically (CLI)

You can enable the device to automatically renew certificates that were acquired by online enrollment or loaded manually. This feature saves you from having to remember to renew certificates on the device before they expire, and it helps to maintain valid certificates at all times.

Automatic certificate renewal is disabled by default. You can configure the device to automatically send out a request to renew a certificate before it expires. You can set the time when you want the device to send out the certificate renewal request in number of days and minutes before the expiration date. By setting different times for each certificate, you prevent the device from having to renew all certificates at the same time.

Before you begin: Obtain a certificate either online or manually. See “Enabling Digital Certificates Online: Configuration Overview” on page 388.

For this feature to work, the device must be able to reach the SCEP server, and the certificate must be present on the device during the renewal process. Furthermore, for this feature to work, you must also ensure that the CA issuing the certificate can return

the same DN. The CA must not modify the subject name and alternate subject name extension in the new certificate.

You can enable and disable automatic SCEP certificate renewal for all SCEP certificates or on a per-certificate basis.

To enable and configure certificate reenrollment use the **set security pki auto-re-enrollment** command with the following information:

- Certificate ID of the CA certificate—for example, **sm1**.
- Name of the CA profile associated with the certificate—for example, **aaa**.
- Challenge password for CA certificate enrollment and revocation. This password must be the same one configured previously for the CA—for example, **abc**.
- Trigger time for the reenrollment. This value sets the certificate reenrollment time as a percentage of the time left before expiration. For example, to start reenrollment when 10 percent of the certificate time remains, specify **10**.
- During automatic reenrollment, by default the Juniper Networks device uses the existing key pair. To generate a new key pair, specify **re-generate-key-pair**.

For example:

```
user@host# set security pki auto-re-enrollment certificate-id sm1 ca-profile-name aaa
challenge-password abc re-enroll-trigger-time-percentage 10 re-generate-keypair
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Example: Checking Certificate Validity Using CRLs (CLI) on page 403

Deleting Certificates (CLI Procedure)

You can delete a local or trusted CA certificate that is automatically or manually generated.

Use the following command to delete a local certificate:

```
user@host> clear security pki local certificate certificate-id (certificate-id | all |
system-generated )
```

Specify a certificate ID to delete a local certificate with a specific ID, use **all** to delete all local certificates, or specify **system-generated** to delete the automatically generated self-signed certificate.

When you delete an automatically generated self-signed certificate, the device generates a new one.

To delete a CA certificate:

```
user@host> clear security pki ca-certificate ca-profile (ca-profile-name | all)
```

Specify a CA profile to delete a specific CA certificate, or use **all** to delete all CA certificates present in the persistent store.



NOTE: You are asked for confirmation before a CA certificate can be deleted.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Example: Verifying Certificate Validity (CLI) on page 402
 - Example: Checking Certificate Validity Using CRLs (CLI) on page 403

Self-Signed Certificates

- Understanding Self-Signed Certificates on page 398
- Using Automatically Generated Self-Signed Certificates (CLI Procedure) on page 399
- Example: Manually Generating Self-Signed Certificates (CLI) on page 400

Understanding Self-Signed Certificates

A self-signed certificate is a certificate that is signed by its creator rather than by a Certificate Authority (CA).

Self-signed certificates allow for use of SSL-based (Secure Sockets Layer) services without requiring that the user or administrator undertake the considerable task of obtaining an identity certificate signed by a CA.



NOTE: Self-signed certificates do not provide additional security as do those generated by CAs. This is because a client cannot verify that the server he or she has connected to is the one advertised in the certificate.

This topic includes the following sections:

- Generating Self-Signed Certificates on page 398
- Automatically Generating Self-Signed Certificates on page 399
- Manually Generating Self-Signed Certificates on page 399

Generating Self-Signed Certificates

Junos OS provides two methods for generating a self-signed certificate:

- Automatic generation

In this case, the creator of the certificate is the Juniper Networks device. An automatically generated self-signed certificate is configured on the device by default.

After the device is initialized, it checks for the presence of an automatically generated self-signed certificate. If it does not find one, the device generates one and saves it in the file system.

- Manual generation

In this case, you create the self-signed certificate for the device.

At any time, you can use the CLI to generate a self-signed certificate. These certificates are also used to gain access to SSL services.

Self-signed certificates are valid for five years from the time they were generated.

Automatically Generating Self-Signed Certificates

An automatically generated self-signed certificate allows for use of SSL-based services without requiring that the administrator obtain an identity certificate signed by a CA.

A self-signed certificate that is automatically generated by the device is similar to a Secure Shell (SSH) host key. It is stored in the file system, not as part of the configuration. It persists when the device is rebooted, and it is preserved when a **request system snapshot** command is issued.

Manually Generating Self-Signed Certificates

A self-signed certificate that you manually generate allows for use of SSL-based services without requiring that you obtain an identity certificate signed by a CA. A manually generated self-signed certificate is one example of a public key infrastructure (PKI) local certificate. As is true of all PKI local certificates, manually generated self-signed certificates are stored in the file system.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Certificates on page 386
 - Using Automatically Generated Self-Signed Certificates (CLI Procedure) on page 399
 - Example: Manually Generating Self-Signed Certificates (CLI) on page 400

Using Automatically Generated Self-Signed Certificates (CLI Procedure)

After the device is initialized, it checks for the presence of a self-signed certificate. If a self-signed certificate is not present, the device automatically generates one.

You can add the following statement to your configuration if you want to use the automatically generated self-signed certificate to provide access to HTTPS services:

```
system {
  services {
    web-management {
      http {
        interface [ ... ];
      } https {
        system-generated-certificate;
        interface [ ... ];
      }
    }
  }
}
```

The device uses the following distinguished name for the automatically generated certificate:

“CN=<device serial number>, CN=system generated, CN=self-signed”

Use the following command to specify that the automatically generated self-signed certificate is to be used for Web management HTTPS services:

```
user@host# set system services web-management https system-generated-certificate
```

Use the following operational command to delete the automatically generated self-signed certificate:

```
user@host# clear security pki local-certificate system-generated
```

After you delete the system-generated self-signed certificate, the device automatically generates a new one and saves it in the file system.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Self-Signed Certificates on page 398
 - Example: Manually Generating Self-Signed Certificates (CLI) on page 400
 - Verifying the Validity of a Certificate: Configuration Overview on page 389

Example: Manually Generating Self-Signed Certificates (CLI)

For a manually generated self-signed certificate, you specify the DN when you create it. (For an automatically generated self-signed certificate, the system supplies the DN, identifying itself as the creator.)

Use the following CLI command to manually generate a self-signed certificate created and signed by the user whose e-mail address is mholmes:

```
user@host# request security pki local-certificate generate-self-signed certificate-id  
self-cert subject cn=abc domain-name Juniper.net ip-address 1.2.3.4 email  
mholmes@juniper.net
```

Use the following CLI command to direct the device to use a manually generated self-signed certificate called self-cert for Web management:

```
user@host# set system services web-management https pki-local-certificate self-cert
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Self-Signed Certificates on page 398
 - Using Automatically Generated Self-Signed Certificates (CLI Procedure) on page 399

Certificate Revocation Lists

- Understanding Certificate Revocation Lists on page 401
- Example: Manually Loading a CRL onto the Device (CLI) on page 401
- Example: Verifying Certificate Validity (CLI) on page 402
- Example: Checking Certificate Validity Using CRLs (CLI) on page 403
- Deleting a Loaded CRL (CLI Procedure) on page 403

Understanding Certificate Revocation Lists

In the normal course of business, certificates are revoked for various reasons. You might wish to revoke a certificate if you suspect that it has been compromised, for example, or when a certificate holder leaves the company.

You can manage certificate revocations and validations in two ways:

- Locally— This is a limited solution.
- By referencing a Certificate Authority (CA) certificate revocation list (CRL). You can automatically access the CRL online at intervals you specify or at the default interval set by the CA.

In Phase 1 negotiations, participants check the CRL list to see if certificates received during an IKE exchange are still valid. If a CRL did not accompany a CA certificate and is not loaded on the device, the device tries to download it automatically from the CRL distribution point of the local certificate. If the device fails to connect to the URL in the certificate distribution point (CDP), it tries to retrieve the CRL from the URL configured in the CA profile.

If the certificate does not contain a certificate distribution point extension, and you cannot automatically retrieve the CRL through Lightweight Directory Access Protocol (LDAP) or Hypertext Transfer Protocol (HTTP), you can retrieve a CRL manually and load that in the device.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Certificates on page 386
 - Example: Checking Certificate Validity Using CRLs (CLI) on page 403
 - Deleting a Loaded CRL (CLI Procedure) on page 403
 - Understanding Public Key Infrastructure on page 383
 - Example: Manually Loading a CRL onto the Device (CLI) on page 401

Example: Manually Loading a CRL onto the Device (CLI)

You can load a CRL manually, or you can have the device load it automatically when you verify certificate validity. To load a CRL manually, you obtain the CRL from a CA and transfer it to the device (for example, using FTP).

Before you begin:

1. Generate a public and private key pair. See “Example: Generating a Public-Private Key Pair (CLI)” on page 390.
2. Generate a certificate request. See “Example: Generating a Local Certificate Request Manually (CLI)” on page 394.

3. Configure a certificate authority (CA) profile. See “Example: Configuring a Certificate Authority Profile (CLI)” on page 391.
4. Load your certificate onto the device. See “Example: Loading CA and Local Certificates Manually (CLI)” on page 395.

With the following command, you load a CRL certificate called **revoke.crl** from the **/var/tmp** directory on the device. The CA profile is called **ca-profile-ipsec**. (Maximum file size is 5 MB.)

```
user@host> request security pki crl load ca-profile ca-profile-ipsec filename  
/var/tmp/revoke.crl
```



NOTE: Junos OS supports loading of CA certificates in X509, PKCS #7, DER, or PEM formats.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Certificate Revocation Lists on page 401
- Digital Certificates Configuration Overview on page 387

Example: Verifying Certificate Validity (CLI)

The CRL is updated automatically, but you must verify certificates manually to find out if a certificate has been revoked, or if the CA certificate used to create a local certificate is no longer present on the device.

When you verify certificates manually, the device uses the CA certificate to verify the local certificate. If the local certificate is valid, and if **revocation-check** is enabled in the CA profile, the device verifies that the CRL is loaded and valid. If not, the device downloads the new CRL.

Use the following command to verify the validity of a local certificate called **local.cert**:

```
user@host> request security pki local-certificate verify certificate-id local.cert
```

Use the following command to verify the validity of a CA certificate called **ca-cert**:

```
user@host> request security pki ca-certificate verify certificate-id ca-cert
```



NOTE: The associated private key and the signature are also verified.

For more information on the certificate, use the **show** commands (**show security pki ca-certificate** and **show security pki certificate-request**) described in the *Junos OS CLI Reference*.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Certificate Revocation Lists on page 401
- Example: Checking Certificate Validity Using CRLs (CLI) on page 403

- Deleting Certificates (CLI Procedure) on page 397

Example: Checking Certificate Validity Using CRLs (CLI)

In Phase 1 negotiations, participants check the CRL list to see if certificates received during an IKE exchange are still valid. If a CRL did not accompany a CA certificate and is not loaded on the device, Junos OS tries to retrieve the CRL through the LDAP or HTTP CRL location defined within the CA certificate itself. If no URL address is defined in the CA certificate, the device uses the URL of the server that you define for that CA certificate. If you do not define a CRL URL for a particular CA certificate, the device gets the CRL from the URL in the CA profile configuration.



NOTE: The CRL distribution point extension (.cdp) in an X509 certificate can be added to either an HTTP URL or an LDAP URL.

With the following command, you direct the device to check the validity of the CA profile called **my_profile** and, if a CRL did not accompany a CA certificate and is not loaded on the device, to retrieve the CRL from the URL **http://abc**.

```
user@host# set security pki ca-profile my_profile revocation-check crl url http://abc
```

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Certificate Revocation Lists on page 401
- Deleting a Loaded CRL (CLI Procedure) on page 403
- Deleting Certificates (CLI Procedure) on page 397

Deleting a Loaded CRL (CLI Procedure)

You can choose to delete a loaded CRL if you no longer need to use it to manage certificate revocations and validation.

Use the following command to delete a loaded certificate revocation list:

```
user@host> clear security pki crl ca-profile (ca-profile | all)
```

Specify a CA profile to delete a CRL associated with the CA identified by the profile, or use **all** to delete all CRLs.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Checking Certificate Validity Using CRLs (CLI) on page 403
- Deleting Certificates (CLI Procedure) on page 397

CHAPTER 20

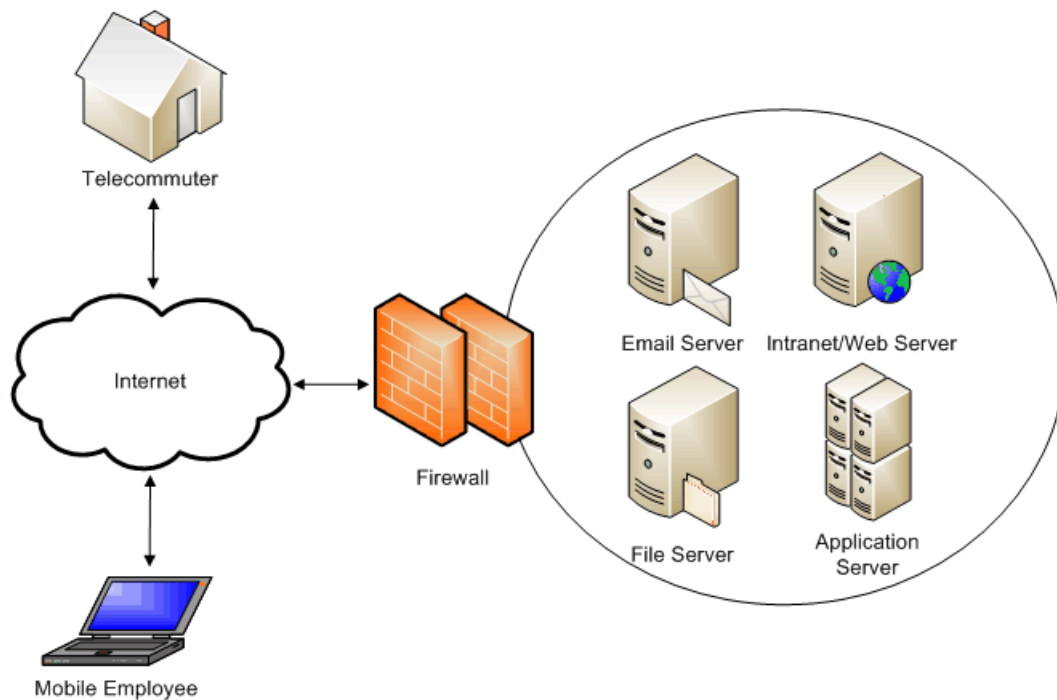
Dynamic VPNs

- [Dynamic VPN Overview on page 405](#)
- [Dynamic VPN Configuration Overview on page 407](#)
- [Dynamic VPN Client Configurations on page 408](#)
- [Dynamic VPN Global Client Download Settings on page 409](#)
- [Dynamic VPN and Access Manager User Experience on page 410](#)
- [Access Manager Client-Side Reference on page 414](#)

Dynamic VPN Overview

Virtual private network (VPN) tunnels enable users to securely access assets such as e-mail servers and application servers that reside behind a firewall. End-to-site VPN tunnels are particularly helpful to remote users such as telecommuters because a single tunnel enables access to all of the resources on a network—the users do not need to configure individual access settings to each application and server. See Figure 53 on page 406.

Figure 53: Using a VPN Tunnel to Enable Remote Access to a Corporate Network



The dynamic VPN feature further simplifies remote access by enabling users to establish Internet Protocol Security (IPsec) VPN tunnels without having to manually configure VPN settings on their PCs or laptops. Instead, authenticated users can simply download the Access Manager Web client to their computers. This Layer 3 remote access client uses client-side configuration settings that it receives from the server to create and manage a secure end-to-site VPN tunnel to the server.



NOTE: The dynamic VPN feature is disabled by default on the device. You must enable and configure it before you can use it.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Dynamic VPN Configuration Overview on page 407
- Understanding Dynamic VPN Client Configurations on page 408
- Understanding Dynamic VPN Global Client Download Settings on page 409
- Understanding the Dynamic VPN and Access Manager User Experience on page 410
- Access Manager Client-Side System Requirements on page 414

Dynamic VPN Configuration Overview

The dynamic VPN feature secures traffic through your network by passing it through IPsec VPN tunnels. To configure an IPsec VPN tunnel, you must specify Phase 1 settings (which enable participants to establish a secure channel in which to negotiate the IPsec security association (SA), and Phase 2 settings (which enable participants to negotiate the IPsec SA that authenticates traffic flowing through the tunnel). This topic describes the order in which you must configure these tunnel negotiation settings as well as other tasks you must complete to enable the tunnels on your network.

To configure the dynamic VPN feature, you must do the following:

1. Define an outgoing interface by using the **interfaces** configuration statement.
Use this interface to pass IKE security associations (SAs) through the device. (You need to select this interface when configuring your IKE gateway.) See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security policies by using the **security policies** configuration statement.
Use these policies to define which traffic can pass through your network. (After you create your VPN configuration, you need to add it to this policy.) See “Example: Configuring a Security Policy to Permit or Deny All Traffic” on page 121.
3. Create at least one access profile by using the **access profile** configuration statement.
Use the access profile(s) to control the authentication of users who want to download Access Manager and users who want to establish dynamic VPN tunnels to your firewall. (You need to select these access profiles when configuring your IKE gateway and dynamic VPN global options. Note that you can use the same access profile to authenticate users in both cases, or you can use separate access profiles to authenticate downloads and VPN sessions.) See:
 - Example: Configuring Pass-Through Authentication on page 299
 - Example: Configuring Web Authentication on page 306
4. Create an IKE gateway to include in your VPN configuration:
 - a. Create one or more IKE Phase 1 proposals by using the **security ike proposal** configuration statement. (You need to select this proposal when configuring your IKE policy.) See “Example: Configuring an IKE Phase 1 Proposal (CLI)” on page 369.
 - b. Create one or more IKE policies by using the **security ike policy** configuration statement. (You need to select this policy when configuring your IKE gateway.) See “Example: Configuring an IKE Policy (CLI)” on page 370.
 - c. Create an IKE gateway configuration by using the **security ike gateway** configuration statement. (You need to select this gateway when configuring your IPsec AutoKey.) See “Example: Configuring an IKE Gateway (CLI)” on page 370.
5. Create an IPsec AutoKey to include in your VPN configuration:

- a. Create one or more IPsec Phase 2 proposals by using the **security ipsec proposal** configuration statement. (You need to select this proposal when configuring your IPsec policy.) See “Example: Configuring an IPsec Phase 2 Proposal (CLI)” on page 373.
 - b. Create one or more IPsec policies by using the **security ipsec policy** configuration statement. (You need to select this policy when configuring your IPsec AutoKey.) See “Example: Configuring an IPsec Policy (CLI)” on page 373.
 - c. Create an IKE AutoKey configuration by using the **security ipsec autokey** configuration statement. (You need to select this IKE AutoKey configuration when configuring your VPN client configuration.) See “Example: Configuring AutoKey IKE (CLI)” on page 374.
6. Create a client VPN configuration by using the **security dynamic-vpn clients** configuration statement.

The settings are downloaded as part of the client to your users' computers and are used to establish the dynamic VPN tunnels between the clients and the server. For more detailed configuration instructions, see “Example: Creating a Dynamic VPN Client Configuration (CLI)” on page 409.
 7. Update your security policy (or policies) to include your client VPN configuration by using the **security from-zone zone-name to-zone zone-name policy then permit tunnel ipsec-vpn vpn-name** configuration statement. See “Example: Configuring a Security Policy to Permit or Deny All Traffic” on page 121.
 8. Specify global settings for client downloads by using the **security dynamic-vpn access-profile** configuration statement and the **security dynamic-vpn force-upgrade** configuration statement. For more detailed configuration instructions, see “Example: Configuring Dynamic VPN Global Client Download Settings (CLI)” on page 410

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Dynamic VPN Overview on page 405

Dynamic VPN Client Configurations

- Understanding Dynamic VPN Client Configurations on page 408
- Example: Creating a Dynamic VPN Client Configuration (CLI) on page 409

Understanding Dynamic VPN Client Configurations

The client configuration controls which resources should be protected by the VPN configuration and specifies which users can download the client. You can create multiple client configurations as part of your setup. Each configuration includes an IKE ID for the user (such as johndoe.yourcompany.com), a Phase 1 security key, and a generated token to establish eligibility for future client downloads.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Dynamic VPN Overview on page 405

- Example: Creating a Dynamic VPN Client Configuration (CLI) on page 409

Example: Creating a Dynamic VPN Client Configuration (CLI)

In this example, you configure the device to protect the 10.100.100.0/24 resource, but specify that the 0.0.0.0/0, 1.1.1.1/24, and 0.0.0.0/32 resources are exempt. You specify that the dynamic VPN feature should use the ipsec-config VPN configuration to create the VPN tunnel. In addition, you add johndoe and janedoe to the list of users who can download the client.

To create a client configuration using the CLI editor:

```
user@host# set security dynamic-vpn clients config1 remote-protected-resources
10.100.100.0/24
user@host# set security dynamic-vpn clients config1 remote-exceptions 0.0.0.0/0,
1.1.1.1/24, 0.0.0.0/32
user@host# set security dynamic-vpn clients config1 ipsec-vpn ipsec-config
user@host# set security dynamic-vpn clients config1 user johndoe, janedoe
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Dynamic VPN Configuration Overview on page 407
 - Understanding Dynamic VPN Client Configurations on page 408

Dynamic VPN Global Client Download Settings

- Understanding Dynamic VPN Global Client Download Settings on page 409
- Example: Configuring Dynamic VPN Global Client Download Settings (CLI) on page 410

Understanding Dynamic VPN Global Client Download Settings

Global dynamic VPN settings enable you to control the following options:

- Access Manager download authentication—Use access profile(s) to control the authentication of users who want to download Access Manager.



NOTE: You must use access profiles to authenticate users who want to download Access Manager and to authenticate users who want to establish dynamic VPN tunnels to your firewall. Note that you can use the same access profile to authenticate users in both cases, or you can use separate access profiles to authenticate downloads and VPN sessions.

- Forced upgrades—Use the force upgrade option to automatically upgrade the client's software when a more recent version is available. If you do not enable this option, the user is given a choice to manually upgrade the client's software when a more recent version is available.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Dynamic VPN Overview on page 405

- Example: Configuring Dynamic VPN Global Client Download Settings (CLI) on page 410

Example: Configuring Dynamic VPN Global Client Download Settings (CLI)

In this example, you configure the device to use the remote-users access profile to control the authentication of users who try to download Access Manager. You also configure the setup program to automatically upgrade the client on users' machines when it detects that a more current version of the client is available on the server.

To configure global dynamic VPN settings using the CLI editor:

```
user@host# set security dynamic-vpn clients remote-users
user@host# set security dynamic-vpn force-upgrade
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Dynamic VPN Configuration Overview on page 407
 - Understanding Dynamic VPN Global Client Download Settings on page 409

Dynamic VPN and Access Manager User Experience

- Understanding the Dynamic VPN and Access Manager User Experience on page 410
- Connecting to the Remote Access Server for the First Time (Pre-IKE Phase) on page 411
- Connecting to the Remote Access Server for Subsequent Sessions (Pre-IKE Phase) on page 412
- Establishing an IPsec VPN Tunnel (IKE Phase) on page 413

Understanding the Dynamic VPN and Access Manager User Experience

From the user's perspective, creating a secure VPN tunnel consists of two simple phases:

1. Connect to the remote access server (Pre-IKE phase).

The first time a user needs to establish a VPN tunnel, they simply navigate to <https://<serverhost>/dynamic-vpn> and enter their username and password in the login page that appears. Assuming that the user authenticates successfully and has administrator privileges, the Juniper Networks device (also called the remote access server) installs Access Manager on the user's computer and provides a VPN configuration that is specific to the user.

2. Establish the VPN tunnel (IKE phase).

The Access Manager client provides the user with a simple GUI for launching the client configuration; the client configuration does all the work of establishing and negotiating the IPsec VPN tunnel for the user. Once installed, Access Manager and the client configuration are available for future IPsec VPN sessions.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Connecting to the Remote Access Server for the First Time (Pre-IKE Phase) on page 411

- Connecting to the Remote Access Server for Subsequent Sessions (Pre-IKE Phase) on page 412
- Establishing an IPsec VPN Tunnel (IKE Phase) on page 413
- Dynamic VPN Overview on page 405

Connecting to the Remote Access Server for the First Time (Pre-IKE Phase)

To establish a secure VPN tunnel from the user's computer to the remote access server, the user must first authenticate into the server and download the client-side files as follows:

1. The user accesses the server's URL.

The user navigates to the <https://<serverhost>/dynamic-vpn> URL through a Web browser. This URL directs the user to the dynamic VPN login page on the remote access server.

2. The user signs into the server.

The user enters the appropriate username and password into the login page, and the remote access server sends them to the authentication server for validation.

3. The server retrieves a client configuration.

Once the server determines that the user has successfully authenticated, the server determines which client configuration to use when creating a secure VPN tunnel. The configuration includes an IKE ID for the user (such as johndoe.yourcompany.com), a Phase 1 security key, and a generated token to establish eligibility for future client downloads.

4. The server downloads the setup client to the user's computer.

The server downloads the setup client to the user's computer. The server downloads the setup client (along with the client version information, client initialization parameters, and client VPN configuration parameters) to the user's computer:

- If the user is using Internet Explorer with Active-X enabled, the remote access server downloads an Active-X setup client to the user's computer.
- Otherwise, if the user is using a Web browser with Java enabled, the remote access server downloads an Java setup client to the user's computer.
- If the user does not have Active-X or Java enabled, the server presents a download page to the user, enabling the user to manually download the setup client.

5. The setup client checks that the user has administrator privileges.

Once the server has successfully downloaded the setup client to the user's computer, the setup client checks that the user has the proper rights to install a new client. (Administrator privileges are required only to install the client, but not to upgrade it.)

6. The setup client installs Access Manager.

The setup client installs Access Manager on the user's computer. The user is prompted to restart the computer to finish the installation.

Once the Access Manager client is successfully launched, the user can initiate a secure VPN connection to the remote access server from Access Manager.



NOTE:

- You can also download the latest version of the Access Manager client from the [Juniper Networks Support site](#).
- The user can connect to the remote access server and initiate a client download before you have finished configuring the dynamic VPN feature. In this case, the user can still authenticate into the server, but cannot establish a secure VPN tunnel.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Dynamic VPN Configuration Overview on page 407
- Understanding the Dynamic VPN and Access Manager User Experience on page 410
- Connecting to the Remote Access Server for Subsequent Sessions (Pre-IKE Phase) on page 412
- Establishing an IPsec VPN Tunnel (IKE Phase) on page 413

Connecting to the Remote Access Server for Subsequent Sessions (Pre-IKE Phase)

If the user has already downloaded Access Manager, the process for signing into the remote access server for subsequent sessions is as follows:

1. The user accesses the server.

The user launches the Access Manager client that is preinstalled on the computer (recommended). Alternatively, the user can access the server through the dynamic VPN URL (<https://<serverhost>/dynamic-vpn>). If the user chooses to use this method, however, the server downloads Access Manager to the user's desktop—even if the client already exists. As part of this process, the remote access server prompts the user for a username and password, checks that the user has the proper install privileges, generates a new connection token, and downloads the setup client.

2. The client determines if an upgrade is required.

The client checks the client configuration version installed on the user's computer. If a more recent version of the client is available, the client code either automatically upgrades the client software (if you have enabled the Force Upgrade option) or gives the user the option of upgrading (if you have not enabled the Force Upgrade option).

The user can initiate a secure VPN connection to the remote access server from Access Manager once it is successfully launched.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Dynamic VPN Configuration Overview on page 407
- Understanding the Dynamic VPN and Access Manager User Experience on page 410
- Connecting to the Remote Access Server for the First Time (Pre-IKE Phase) on page 411
- Establishing an IPsec VPN Tunnel (IKE Phase) on page 413

Establishing an IPsec VPN Tunnel (IKE Phase)

Once Access Manager is installed, the user can use it to initiate a secure VPN tunnel to the remote access server as follows:

1. The user launches Access Manager.

The user can launch Access Manager by using either of the following methods:

- Choose **All Programs > Juniper Networks > Access Manager** from the Windows Start menu.
- Select the Access Manager icon in the system tray at the lower right corner of the Windows screen.

When the user launches the client, the Access Manager dialog box appears.

2. The user creates a connection to the server, if necessary.

If no connections are available in the Access Manager dialog box, the user must specify a connection server:

- a. From the File menu, choose **Setup Connection**.
- b. In the New Connection dialog box that appears, enter the hostname of the remote access server and the appropriate username.
- c. Click **OK**. The specified connection appears in the Access Manager dialog box.

3. The user starts the connection.

In the Access Manager dialog box, the user selects which server connection to initiate by using one of the following methods:

- Select one of the connections, right-click, and choose **Connect**.
- Select one of the connections, and from the File menu, choose **Start Connection**.

4. The server checks for a valid license.

When the user initiates a connection to a remote access server, the server checks that a seat license is currently available for the user's session.

5. The user signs into the server.

The user enters the appropriate username and password into the login page, and the remote access server sends them to the authentication server for validation.



NOTE: The username and password entered here are used to validate the user's eligibility to establish the VPN session. These credentials are separate from those used to validate the user's eligibility to download the client.

6. The client initiates the VPN session.

Once the user has successfully authenticated, the client sends a preshared key to the remote access server. (The client initially received this key as part of the initial client configuration download.) The client and server use an AutoKey IKE exchange to create security associations (SAs) and establish a secure VPN tunnel.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Dynamic VPN Configuration Overview on page 407
 - Understanding the Dynamic VPN and Access Manager User Experience on page 410
 - Connecting to the Remote Access Server for the First Time (Pre-IKE Phase) on page 411
 - Connecting to the Remote Access Server for Subsequent Sessions (Pre-IKE Phase) on page 412

Access Manager Client-Side Reference

- Access Manager Client-Side System Requirements on page 414
- Access Manager Client-Side Files on page 415
- Access Manager Client-Side Registry Changes on page 417
- Access Manager Client-Side Error Messages on page 418
- Troubleshooting Access Manager Client-Side Problems on page 421

Access Manager Client-Side System Requirements

The user can install Access Manager on any Windows XP or Windows Vista machine with an Internet connection. The user must have administrator privileges to install the client, but not to run it.

Access Manager can run simultaneously on the same computer with other Juniper Networks clients, including the Odyssey Access Client (OAC), Network Connect client, Windows Secure Application Manager (WSAM) client, Host Checker client, and WX client.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding the Dynamic VPN and Access Manager User Experience on page 410
 - Access Manager Client-Side Files on page 415
 - Access Manager Client-Side Registry Changes on page 417
 - Access Manager Client-Side Error Messages on page 418
 - Troubleshooting Access Manager Client-Side Problems on page 421

Access Manager Client-Side Files

Table 38 on page 415 lists the directories where Access Manager installs files on a user's computer, the files it installs, and the files that remain after the user uninstalls the client.

Table 38: Access Manager Client-Side Files

Installation Directory	Files Installed in Directory	Files Remaining After Uninstall
%COMMONFILES%\Juniper Networks\Connection Manager	<ul style="list-style-type: none"> • ConnectionManagerService.dll • install.log • Uninstall.exe • Uninstall.exe.manifest • versionInfo.ini 	install.log
%COMMONFILES%\Juniper Networks\ConnectionStore	<ul style="list-style-type: none"> • ConnectionStoreService.dll • dcfDOM.dll • install.log • Uninstall.exe • Uninstall.exe.manifest • versionInfo.ini 	install.log
%COMMONFILES%\Juniper Networks\IPSecMgr	<ul style="list-style-type: none"> • install.log • ipsecmgr.dll • Uninstall.exe • Uninstall.exe.manifest • versionInfo.ini 	install.log
PROGRAMFILES%\Juniper Networks\Juniper Access Manager	<ul style="list-style-type: none"> • AccessServiceComponent.x86.exe • ConnectionMgrComponent.x86.exe • ConnectionStoreComponent.x86.exe • install.log • IPSecMgrComponent.x86.exe • JamGUIComponent.x86.exe • JamInstaller.dep • jnprnaInstall.exe • TunnelManagerComponent.x86.exe • Uninstall.exe • Uninstall.exe.manifest • versionInfo.ini • vpnAccessMethodComponent.x86.exe 	install.log Log file location: C:\Documents and Settings\All Users\Application Data\Juniper Networks\Logging

Table 38: Access Manager Client-Side Files (*continued*)

Installation Directory	Files Installed in Directory	Files Remaining After Uninstall
%COMMONFILES%\Juniper Networks\JamUI	<ul style="list-style-type: none"> install.log jamCommand.exe jamTray.exe jamUI.exe jamUIResource_EN.dll uiPlugin.dll Uninstall.exe Uninstall.exe.manifest versionInfo.ini 	install.log
%COMMONFILES%\Juniper Networks\JUNS	<ul style="list-style-type: none"> access.ini dsAccessService.exe dsInstallerService.dll dsLogService.dll install.log Uninstall.exe Uninstall.exe.manifest versionInfo.ini 	install.log
%COMMONFILES%\Juniper Networks\JNPRNA	<ul style="list-style-type: none"> install.log jnpna.cat jnpna.inf jnpna.sys jnpnaapi.dll jnpnaNetInstall.dll jnpnaNetInstall.log jnpna_m.cat jnpna_m.inf jnpna.cat jnpna.inf jnpna.sys jnpnamgr.cat jnpnamgr.dll jnpnamgr.inf jnpnamgr.sys nsStatsDump.exe uninst.exe versionInfo.ini %WINDIR%\system32\drivers\jnpna.sys %WINDIR%\system32\drivers\jnpna.sys %WINDIR%\system32\drivers\jnpnamgr.sys 	<ul style="list-style-type: none"> install.log jnpnaNetInstall.log

Table 38: Access Manager Client-Side Files (*continued*)

Installation Directory	Files Installed in Directory	Files Remaining After Uninstall
COMMONFILES%\Juniper Networks\Tunnel Manager	<ul style="list-style-type: none"> dsTMClient.dll dsTMService.dll dsTunnelManager.dll install.log TM.dep Uninstall.exe Uninstall.exe.manifest versionInfo.ini 	install.log
%COMMONFILES%\Juniper Networks\vpnAccessMethod	<ul style="list-style-type: none"> install.log Uninstall.exe Uninstall.exe.manifest versionInfo.ini vpnAccessMethod.dll vpnAccessMethod_EN.dll 	install.log

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding the Dynamic VPN and Access Manager User Experience on page 410
 - Access Manager Client-Side System Requirements on page 414
 - Access Manager Client-Side Registry Changes on page 417
 - Access Manager Client-Side Error Messages on page 418
 - Troubleshooting Access Manager Client-Side Problems on page 421

Access Manager Client-Side Registry Changes

Table 39 on page 418 lists the Windows Registry changes that the Access Manager client and components make to your users' computers when creating dynamic VPN tunnels.

Table 39: Access Manager Client-Side Registry Changes

Registry Key Location	Registry Key Changes
HKEY_LOCAL_MACHINE\SOFTWARE\Juniper Networks\Common Files	<ul style="list-style-type: none"> • jnprnaapi="C:\\Program Files\\Common Files\\Juniper Networks\\JNPRNA\\jnprnaapi.dll • jnprvamgr="C:\\Program Files\\Common Files\\Juniper Networks\\JNPRNA\\jnprvamgr.dll • nsStatsDump="C:\\Program Files\\Common Files\\Juniper Networks\\JNPRNA\\nsStatsDump.exe • dsLogService="C:\\Program Files\\Common Files\\Juniper Networks\\JUNS\\dsLogService.dll • dsTMClient="C:\\Program Files\\Common Files\\Juniper Networks\\Tunnel Manager\\dsTMClient.dll • dsTunnelManager="C:\\Program Files\\Common Files\\Juniper Networks\\Tunnel Manager\\dsTunnelManager.dll
HKEY_LOCAL_MACHINE\SOFTWARE\Juniper Networks\Logging	<ul style="list-style-type: none"> • LogFileName="C:\\Documents and Settings\\All Users\\Application Data\\Juniper Networks\\Logging\\debuglog.log • "Level"="3" • "LogSizeInMB"="10"
HKCU\Software\Juniper Networks\Access Manager\	(Content varies. Contains client configuration data downloaded from the server.)

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding the Dynamic VPN and Access Manager User Experience on page 410
 - Access Manager Client-Side System Requirements on page 414
 - Access Manager Client-Side Files on page 415
 - Access Manager Client-Side Error Messages on page 418
 - Troubleshooting Access Manager Client-Side Problems on page 421

Access Manager Client-Side Error Messages

Table 40 on page 418 lists possible errors that end users might see when installing or running Access Manager, the possible causes for the messages, and suggested actions.

Table 40: Dynamic VPN Client-Side Errors

Error Message	Possible Causes	Suggested User Action
Component instance already in use	Internal error.	Try to reconnect to the firewall.
Memory allocation failure	Internal error.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.

Table 40: Dynamic VPN Client-Side Errors (*continued*)

Error Message	Possible Causes	Suggested User Action
Failed to load connection store	Internal error.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Cannot get connection information for firewall	Internal error. Could not retrieve connection information for the specified firewall.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Authentication failure: Unknown HTTP response code	Internal error. Could not decipher the HTTP response.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Authentication failure: Incorrect username or password	The user entered an invalid username or password.	Reenter your credentials.
Authentication failure: Firewall is out of licenses	All available licenses are currently being used for other dynamic VPN sessions or no licenses are installed for the feature.	Try to reconnect to the firewall once a license has been freed by another user. If the problem persists, contact your system administrator.
Authentication failure: No configuration available	No configuration is currently available for the specified user account.	Contact your system administrator.
Cannot create IPsec route entry	Internal error. Failed to read route entry from the connection store.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Failed to read route entry from connection store	Internal error. Failed to read the route entry from the connection store.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Failed to add route entry to policy	Internal error. Failed to add the route entry to the connection store.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Failed to initialize IPsec Manager	Internal error. Failed to initialize the IPsec Manager.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
IPsec authentication failed	Phase 1 negotiations, extended authentication (XAuth), or Phase 2 negotiations failed.	Try to reconnect to the firewall.
IPsec configuration failed	Internal error or policy configuration error. The Tunnel Manager was unable to configure the local IP settings.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
IKE negotiations failed	The components cannot agree on security parameters during the IKE exchange. The administrator probably needs to reconfigure the Phase 1 proposal.	Contact your system administrator.
Failed to initialize authentication	Failed to authenticate when connecting to the firewall, possibly because the specified hostname did not resolve against the distinguished name server (DNS).	Try to reconnect to the firewall.

Table 40: Dynamic VPN Client-Side Errors (*continued*)

Error Message	Possible Causes	Suggested User Action
Failed to connect to server	The TCP connection to the webserver failed during authentication, possibly because of network connectivity issues.	Try to reconnect to the firewall.
Failed to send initial HTTP request	Webserver authentication failed, possibly because of network connectivity issues.	Try to reconnect to the firewall.
Failed to get HTTP response	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Firewall refused authentication request	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Client failed to provide login page	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Server failed to send authentication request	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Server failed to respond to authentication request	The client sent the user's credentials to the webserver, but the server failed to respond in a useful manner.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Authentication negotiation failed	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
Failed to get configuration from firewall	Webserver authentication failed.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.
The user cancelled authentication.	User canceled authentication	Try to reconnect to the firewall and reenter your credentials.
Failed to enter username or password	Authentication request timed out.	Try to reconnect to the firewall and reenter your credentials.
Server failed to request username and password	The client failed to display the user interface asking the user for credentials.	Exit and restart Access Manager. If the problem persists, contact your system administrator.
Your client state is preventing the connection	The user's client is in an inoperable state.	Try to reconnect to the firewall. If the problem persists, exit and restart Access Manager.
Cannot open connection store	The client could not contact the connection store.	Exit and restart Access Manager. If the problem persists, reinstall Access Manager.
Cannot process configuration provided by firewall	The script provided by the firewall was in some way unusable. The configuration might need to be updated on the server.	Try to reconnect to the firewall. If the problem persists, contact your system administrator.

Table 40: Dynamic VPN Client-Side Errors (*continued*)

Error Message	Possible Causes	Suggested User Action
Access Manager is not running	The Access Manager service is not running.	Exit and restart Access Manager.
Please select a connection	The user chose Start Connection without selecting a connection first.	Select the firewall you want to connect to and then choose Start Connection .
Are you sure you want to delete the selected connection?	The user chose Delete Connection .	Specify whether or not you want to delete the selection connection profile.
Cannot add new connection. Service is not running.	The Access Manager service is not running; therefore it cannot create a new connection profile.	Exit and restart Access Manager. If the problem persists, reinstall Access Manager.
Cannot add new connection	The Access Manager failed to add the new connection profile.	Try again. If the problem persists, exit and restart Access Manager.
Connection name is already in use	Unable to add connection profile because the specified connection name already exists.	Specify a unique name for the connection profile.
Please reinstall Access Manager	Files could not be found when trying to finish the operation.	Reinstall Access Manager.
Invalid server certificate	Certificate validation failed.	Check client-side logs to determine why the certificate failed.
Initializing service...	Initializing one of the client's core components. If the component does not initialize, the client cannot function.	Wait for the service to finish initializing.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding the Dynamic VPN and Access Manager User Experience on page 410
 - Access Manager Client-Side System Requirements on page 414
 - Access Manager Client-Side Files on page 415
 - Access Manager Client-Side Registry Changes on page 417
 - Troubleshooting Access Manager Client-Side Problems on page 421

Troubleshooting Access Manager Client-Side Problems

Problem Users are having problems connecting to the remote access server using Access Manager.

Solution Use the following tools to troubleshoot client-side issues:

- Client-side logs—To view client-side logs, open Access Manager and choose **Save logs and diagnostics** from the File menu. Select a location on your computer to save the zipped log files and click **Save**.

- Detailed logs—To create more detailed client-side logs, open Access Manager and choose **Enable Detailed Logging** from the File menu.
- Firewall connection information—To view connection information for a given firewall, open Access Manager, select the firewall, right-click, and choose **Status**.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding the Dynamic VPN and Access Manager User Experience on page 410
- Access Manager Client-Side System Requirements on page 414
- Access Manager Client-Side Files on page 415
- Access Manager Client-Side Registry Changes on page 417
- Access Manager Client-Side Error Messages on page 418

CHAPTER 21

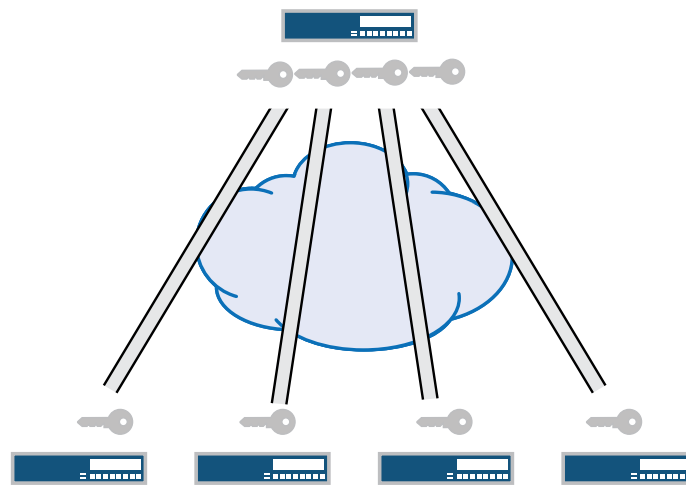
Group VPNs

- Group VPN Overview on page 423
- Understanding the GDOI Protocol on page 425
- Understanding Group Servers and Members on page 426
- Understanding Dynamic Policies on page 427
- Group Key Operations on page 428
- Group VPN Configuration Overview on page 431
- Example: Configuring Group VPN (CLI) on page 432
- Understanding Colocation Mode on page 444
- Example: Configuring Group VPN with Server-Member Colocation (CLI) on page 445
- Understanding IKE Phase 1 Configuration for Group VPN on page 450
- Understanding IPsec SA Configuration for Group VPN on page 451
- Understanding VPN Group Configuration on page 452
- Understanding Antireplay on page 453
- Understanding Server-Member Communication on page 453
- Example: Configuring Server-Member Communication for Unicast Rekey Messages on page 454
- Example: Configuring Server-Member Communication for Multicast Rekey Messages on page 456
- Understanding Heartbeat Messages on page 457
- Understanding Group VPN Limitations on page 458
- Understanding Interoperability with Cisco GET VPN on page 458

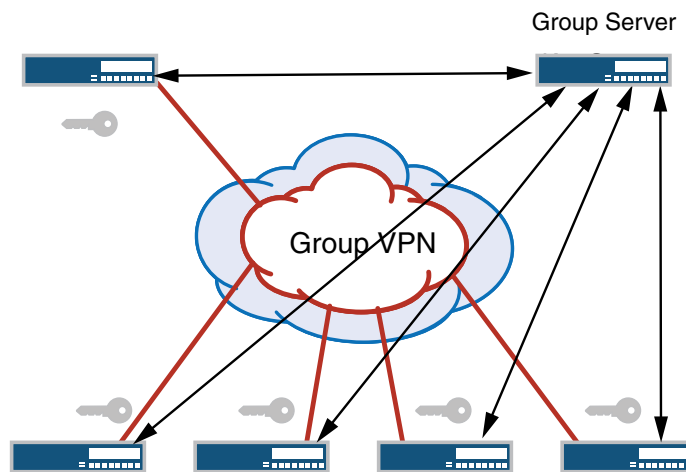
Group VPN Overview

An IPsec security association (SA) is a unidirectional agreement between virtual private network (VPN) participants that defines the rules to use for authentication and encryption algorithms, key exchange mechanisms, and secure communications. With current VPN implementations, the SA is a point-to-point tunnel between two security devices. A group VPN extends IPsec architecture to support SAs that are shared by a group of security devices (see Figure 54 on page 424).

Figure 54: Standard IPsec VPN and Group VPN



Standard IPsec VPN



Group VPN

Server distributes IPsec SA. All members that belong to the group share the same IPsec SA.

With group VPNs, any-to-any connectivity is achieved by preserving the original source and destination IP addresses in the outer header. Secure multicast packets are replicated in the same way as cleartext multicast packets in the core network.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - VPN Overview on page 355
 - Understanding IKE and IPsec Packet Processing on page 361
 - Understanding the GDOI Protocol on page 425
 - Understanding Group Servers and Members on page 426
 - Group VPN Configuration Overview on page 431

Understanding the GDOI Protocol

Group VPN is based on RFC 3547, *The Group Domain of Interpretation (GDOI)*. This RFC describes the protocol between group members and a group server to establish SAs among group members. GDOI messages create, maintain, or delete SAs for a group of devices. The GDOI protocol runs on port 848.

The Internet Security Association and Key Management Protocol (ISAKMP) defines two negotiation phases to establish SAs for an AutoKey IKE IPsec tunnel. Phase 1 allows two devices to establish an ISAKMP SA. Phase 2 establishes SAs for other security protocols, such as GDOI.

With group VPN, Phase 1 ISAKMP SA negotiation is performed between a group server and a group member. The server and member must use the same ISAKMP policy. In Phase 2, GDOI exchanges between the server and member establish the SAs that are shared with other group members. A group member does not need to negotiate IPsec with other group members. GDOI exchanges in Phase 2 must be protected by ISAKMP Phase 1 SAs.

There are two types of GDOI exchanges:

- The **groupkey-pull** exchange allows a member to request SAs and keys shared by the group from the server.
- The **groupkey-push** exchange is a single rekey message that allows the server to send group SAs and keys to members before existing group SAs expire. Rekey messages are unsolicited messages sent from the server to members.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Group VPN Overview on page 423
 - Understanding IKE and IPsec Packet Processing on page 361
 - Understanding Group Servers and Members on page 426
 - Understanding Group Keys on page 428
 - Understanding Rekey Messages on page 429

- Understanding Member Reregistration on page 430
- Understanding Key Activation on page 431

Understanding Group Servers and Members

The center of a group VPN is the group server. The group server performs the following tasks:

- Controls group membership
- Generates encryption keys
- Manages group SAs and keys and distributes them to group members

Group members encrypt traffic based on the group SAs and keys provided by the group server.

A group server can service multiple groups. A single security device can be a member of multiple groups.

Each group is represented by a group identifier, which is a number between 1 and 65,535. The group server and group members are linked together by the group identifier. There can be only one group identifier per group, and multiple groups cannot use the same group identifier.

The following is a high-level view of group VPN server and member actions:

1. The group server listens on UDP port 848 for members to register. A member device must provide correct IKE Phase 1 authentication to join the group. Preshared key authentication on a per-member basis is supported.
2. Upon successful authentication and registration, the member device retrieves group SAs and keys from the server with a GDOI **groupkey-pull** exchange.
3. The server adds the member to the membership for the group.
4. Group members exchange packets encrypted with group SA keys.

The server periodically sends SA and key refreshes to group members with rekey (GDOI **groupkey-push**) messages. Rekey messages are sent before SAs expire; this ensures that valid keys are available for encrypting traffic between group members.

The server also sends rekey messages to provide new keys to members when there is a change in group membership or when the group SA has changed.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Group VPN Overview on page 423
 - Understanding the GDOI Protocol on page 425
 - Understanding Colocation Mode on page 444
 - Understanding Dynamic Policies on page 427

- Understanding Antireplay on page 453
- Group VPN Configuration Overview on page 431

Understanding Dynamic Policies

The group server distributes group SAs and keys to members of a specified group. All members that belong to the same group can share the same set of IPsec SAs. But not all SAs configured for a group are installed on every group member. The SA installed on a specific member is determined by the policy associated with the group SA and the security policies configured on the member.

In a VPN group, each group SA and key that the server pushes to a member is associated with a *group policy*. The group policy describes the traffic on which the key should be used, including protocol, source address, source port, destination address, and destination port.



NOTE: Group policies that are identical (configured with the same source address, destination address, source port, destination port, and protocol values) cannot exist for a single group. An error is returned if you attempt to commit a configuration that contains identical group policies for a group. If this is the case, you must delete one of the identical group policies.

On a group member, a *scope policy* must be configured that defines the scope of the group policy downloaded from the server. A group policy distributed from the server is compared against the scope policies configured on the member. For a group policy to be installed on the member, the following conditions must be met:

- Any addresses specified in the group policy must be within the range of addresses specified in the scope policy.
- The source port, destination port, and protocol specified in the group policy must match those configured in the scope policy.

A group policy that is installed on a member is called a *dynamic policy*.

A scope policy can be part of an ordered list of security policies for a specific from-zone and to-zone context. Junos OS performs a security policy lookup on incoming packets starting from the top of the ordered list.

Depending on the position of the scope policy within the ordered list of security policies, there are several possibilities for dynamic policy lookup:

- If an incoming packet matches a scope policy, the search process continues for a matching dynamic policy. If there is a matching dynamic policy, that policy action (permit) is performed. If there is no matching dynamic policy, then the packet is dropped.



NOTE: In this release, only the tunnel action is allowed for a scope policy. Other actions are not supported.

- If the incoming packet matches a security policy before the scope policy is considered, dynamic policy lookup does not occur.

You configure a scope policy on a group member by using the **policies** configuration statement at the **[edit security]** hierarchy. Use the **ipsec-group-vpn** configuration statement in the permit tunnel rule to reference the group VPN; this allows group members to share a single SA.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Security Policies Overview on page 115
 - Understanding Security Policy Ordering on page 129
 - Example: Configuring a Security Policy to Permit or Deny All Traffic on page 121
 - Understanding the GDOI Protocol on page 425
 - Understanding Group Servers and Members on page 426
 - Group VPN Configuration Overview on page 431

Group Key Operations

This section contains the following topics:

- Understanding Group Keys on page 428
- Understanding Rekey Messages on page 429
- Understanding Member Reregistration on page 430
- Understanding Key Activation on page 431

Understanding Group Keys

The group server maintains a database to track the relationship among VPN groups, group members, and group keys. There are two kinds of group keys that the server downloads to members:

- Key Encryption Key (KEK)—Used to encrypt rekey messages. One KEK is supported per group.
- Traffic Encryption Key (TEK)—Used to encrypt and decrypt IPsec data traffic between group members.

The key associated with an SA is accepted by a group member only if there is a matching scope policy configured on the member. An accepted key is installed for the group VPN, whereas a rejected key is discarded.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Group VPN Overview on page 423
- Understanding the GDOI Protocol on page 425
- Understanding Group Servers and Members on page 426
- Understanding Dynamic Policies on page 427
- Group VPN Configuration Overview on page 431
- Understanding Rekey Messages on page 429
- Understanding Member Reregistration on page 430

Understanding Rekey Messages

If the group is configured for server-member communications (see “Understanding Server-Member Communication” on page 453), the server periodically sends SA and key refreshes to group members with rekey (GDOI **groupkey-push**) messages. Rekey messages are sent before SAs expire; this ensures that valid keys are available for encrypting traffic between group members.

The server also sends rekey messages to provide new keys to members when there is a change in group membership or the group SA has changed (for example, a group policy is added or deleted).

Server-member communications options must be configured on the server to allow the server to send rekey messages to group members. These options specify the type of message and the intervals at which the messages are sent, as explained in the following sections:

- Types of Rekey Messages on page 429
- Rekey Intervals on page 430

Types of Rekey Messages

There are two types of rekey messages:

- Unicast rekey messages—The group server sends one copy of the rekey message to each group member. Upon receipt of the rekey message, members must send an acknowledgment (ACK) to the server. If the server does not receive an ACK from a member (including retransmission of rekey messages), the server considers the member to be inactive and removes it from the membership list. The server stops sending rekey messages to the member.

The **number-of-retransmission** and **retransmission-period** configuration statements for server-member communications control the resending of rekey messages by the server when no ACK is received from a member.

- Multicast rekey messages—The group server sends one copy of the rekey message from the specified outgoing interface to the configured multicast group address. Members do not send acknowledgment of receipt of multicast rekey messages. The registered membership list does not necessarily represent active members because

members might drop out after initial registration. All members of the group must be configured to support multicast messages.



NOTE: IP multicast protocols must be configured to allow delivery of multicast traffic in the network. For detailed information about configuring multicast protocols on Juniper Networks devices, see the *Junos Multicast Protocols Configuration Guide*.

Rekey Intervals

The interval at which the server sends rekey messages is calculated based on the values of the **lifetime-seconds** and **activation-time-delay** configuration statements at the [edit security group-vpn server group] hierarchy. The interval is calculated as **lifetime-seconds** minus 4*(**activation-time-delay**).

The **lifetime-seconds** for the KEK is configured as part of the server-member communications; the default is 3600 seconds. The **lifetime-seconds** for the TEK is configured for the IPsec proposal; the default is 3600 seconds. The **activation-time-delay** is configured for the group on the server; the default is 15 seconds. Using the default values for **lifetime-seconds** and **activation-time-delay**, the interval at which the server sends rekey messages is 3600 minus 4*15, or 3540 seconds.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding the GDOI Protocol on page 425
 - Understanding Group Servers and Members on page 426
 - Understanding Group Keys on page 428
 - Understanding Key Activation on page 431
 - Understanding Member Reregistration on page 430
 - Group VPN Configuration Overview on page 431

Understanding Member Reregistration

If a group member does not receive a new SA key from the server before the current key expires, the member must reregister with the server and obtain updated keys with a GDOI **groupkey-pull** exchange. In this case, the interval at which the server sends rekey messages is calculated as follows: **lifetime-seconds** minus 3*(**activation-time-delay**). Using the default values for **lifetime-seconds** and **activation-time-delay**, the interval at which the server sends rekey messages is 3600 minus 3*15, or 3555 seconds.

Member reregistration can occur for the following reasons:

- The member detects a server reboot by the absence of heartbeats received from the server.
- The rekey message from the group server is lost or delayed, and the TEK lifetime has expired.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding the GDOI Protocol on page 425
 - Understanding Group Servers and Members on page 426
 - Understanding Group Keys on page 428
 - Understanding Rekey Messages on page 429
 - Understanding Key Activation on page 431
 - Group VPN Configuration Overview on page 431

Understanding Key Activation

When a member receives a new key from the server, it waits a period of time before using the key for encryption. This period of time is determined by the **activation-time-delay** configuration statement and whether the key is received through a rekey message sent from the server or as a result of the member reregistering with the server.

If the key is received through a rekey message sent from the server, the member waits $2 * (\text{activation-time-delay})$ seconds before using the key. If the key is received through member reregistration, the member waits the number of seconds specified by the **activation-time-delay** value.

A member retains the two most recent keys sent from the server for each group SA installed on the member. Both keys can be used for decryption, while the most recent key is used for encryption. The previous key is removed the number of seconds specified by the **activation-time-delay** value after the new key is activated.

The default for the **activation-time-delay** configuration statement is 15 seconds. Setting this time period too small can result in a packet being dropped at a remote group member before the new key is installed. Consider the network topology and system transport delays when you change the **activation-time-delay** value. For unicast transmissions, the system transport delay is proportional to the number of group members.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding the GDOI Protocol on page 425
 - Understanding Group Servers and Members on page 426
 - Understanding Group Keys on page 428
 - Understanding Rekey Messages on page 429
 - Understanding Member Reregistration on page 430
 - Group VPN Configuration Overview on page 431

Group VPN Configuration Overview

This topic describes the main tasks for configuring group VPN.

On the group server, configure the following:

1. IKE Phase 1 negotiation. See “Understanding IKE Phase 1 Configuration for Group VPN” on page 450.
2. Phase 2 IPsec SA. See “Understanding IPsec SA Configuration for Group VPN” on page 451.
3. VPN group. See “Understanding VPN Group Configuration” on page 452.

On the group member, configure the following:

1. IKE Phase 1 negotiation. See “Understanding IKE Phase 1 Configuration for Group VPN” on page 450.
2. Phase 2 IPsec SA. See “Understanding IPsec SA Configuration for Group VPN” on page 451.
3. Scope policy that determines which group policies are installed on the member. See “Understanding Dynamic Policies” on page 427.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Group Servers and Members on page 426
 - Understanding Server-Member Communication on page 453
 - Example: Configuring Group VPN (CLI) on page 432
 - Example: Configuring Group VPN with Server-Member Colocation (CLI) on page 445

Example: Configuring Group VPN (CLI)

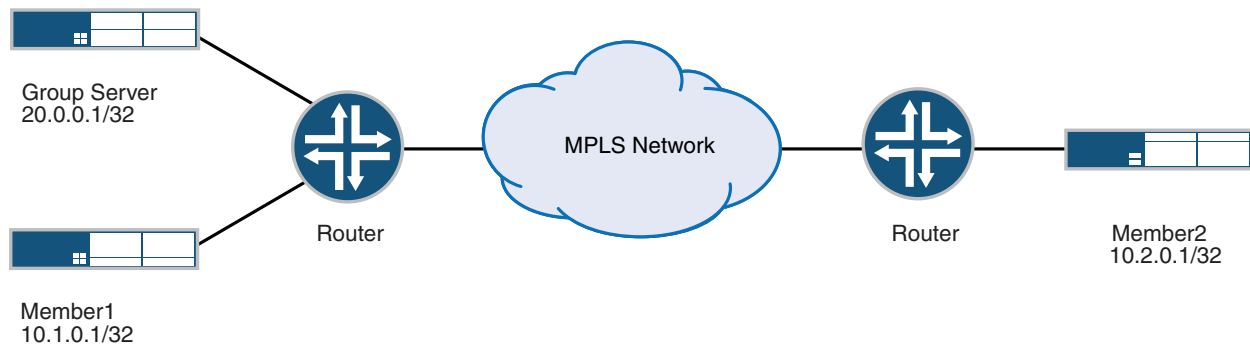
The configuration instructions in this topic describe how to configure a group VPN on server and member devices.

- Overview on page 432
- Configuring the Group Server on page 433
- Configuring Member1 on page 436
- Configuring Member2 on page 438
- Viewing Dynamic Policies on page 442

Overview

In Figure 55 on page 433, a group VPN consists of two member devices (member1 and member2) and a group server (the IP address of the loopback interface on the server is 20.0.0.1). The group identifier is 1.

Figure 55: Server-Member Configuration Example



The Phase 2 group VPN SAs must be protected by a Phase 1 SA. Therefore, group VPN configuration must include configuring IKE Phase 1 negotiations on both the group server and the group members. In addition, the same group identifier must be configured on both the group server and the group member.

Group policies are configured on the group server. All group policies configured for a group are downloaded to group members. Scope policies configured on a group member determine which group policies are actually installed on the member. In this example, the following group policies are configured on the group server for downloading to all group members:

- p1—Allows all traffic from 10.1.0.0/16 to 10.2.0.0/16
- p2—Allows all traffic from 10.2.0.0/16 to 10.1.0.0/16
- p3—Allows multicast traffic from 10.1.1.1/32

The member1 device is configured with scope policies that allow all unicast traffic to and from the 10.0.0.0/8 subnetwork. There is no scope policy configured on member1 to allow multicast traffic; therefore, the SA policy p3 is not installed on member1.

The member2 device is configured with scope policies that drop traffic from 10.1.0.0/16 from the trust zone to the untrust zone and to 10.1.0.0/16 from the untrust zone to the trust zone. Therefore the SA policy p2 is not installed on member2.

Before you begin:

1. Configure the Juniper Networks security devices for network communication.
2. Configure network interfaces on server and member devices. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

Configuring the Group Server

On the group server device, configure the following:

1. Configure the loopback address on the device.

```
[edit]
user@host# edit interfaces
[edit interfaces]
```

```
user@host# set lo0 unit 0 family inet address 20.0.0.1/32
```

2. Configure IKE Phase 1 SA (this configuration must match the Phase 1 SA configured on the group members).

```
[edit security group-vpn server ike]
user@host# set proposal srv-prop authentication-method pre-shared-keys dh-group
group2 authentication-algorithm sha1 encryption-algorithm 3des-cbc
user@host# set policy srv-pol mode main proposals srv-prop pre-shared-key ascii-text
"$9$c1grK8-VYZUHX7UHqmF3Sre"
user@host# set gateway gw1 ike-policy srv-pol address 10.1.0.1
user@host# set gateway gw2 ike-policy srv-pol address 10.2.0.1
```

3. Configure the Phase 2 SA exchange.

```
[edit security group-vpn server ipsec]
user@host# set proposal group-prop authentication-algorithm hmac-sha1-96
encryption-algorithm 3des-cbc lifetime-seconds 3600
```

4. Configure the group identifier, IKE gateway, antireplay time, and server address.

```
[edit security group-vpn server group grp1]
user@host# set group-id 1 ike-gateway srv-gw anti-replay-time-window 120
server-address 20.0.0.1
```

5. Configure server-to-member communications.

```
[edit security group-vpn server group grp1]
user@host# set server-member-communication communication-type unicast
encryption-algorithm aes-128-cbc sig-hash-algorithm md5 certificate "srv-cert"
```

6. Configure the group policies to be downloaded to group members.

```
[edit security group-vpn server group grp1 ipsec-sa group-sa proposal group-prop]
user@host# set match-policy p1 source 10.1.0.0/16 destination 10.2.0.0/16 source-port
0 destination-port 0 protocol 0
user@host# set match-policy p2 source 10.2.0.0/16 destination 10.1.0.0/16
source-port 0 destination-port 0 protocol 0
user@host# set match-policy p3 source 10.1.1.1/16 destination 239.1.1.1/32 source-port
0 destination-port 0 protocol 0
```

7. Confirm your configuration by entering the **show security group-vpn server** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security group-vpn server
ike {
  proposal srv-prop {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
  }
  policy srv-pol {
    mode main;
    proposals srv-prop;
    pre-shared-key ascii-text "$9$gfJUHf5FnCu"; ## SECRET-DATA
  }
  gateway gw1 {
```

```

    ike-policy srv-pol;
    address 10.1.0.1;
  }
  gateway gw2 {
    ike-policy srv-pol;
    address 10.2.0.1;
  }
}
ipsec {
  proposal group-prop {
    authentication-algorithm hmac-sha1-96;
    encryption-algorithm 3des-cbc;
    lifetime-seconds 3600;
  }
}
group grp1 {
  group-id 1;
  anti-replay-time-window 120;
  ike-gateway srv-gw;
  server-address 20.0.0.1;
  server-member-communication {
    communication-type unicast;
    encryption-algorithm aes-128-cbc;
    sig-hash-algorithm md5;
    certificate srv-cert;
  }
}
ipsec-sa group-sa {
  proposal group-prop;
  match-policy p1 {
    source 10.1.0.0/16;
    destination 10.2.0.0/16;
    source-port 0;
    destination-port 0;
    protocol 0;
  }
  match-policy p2 {
    source 10.2.0.0/16;
    destination 10.1.0.0/16;
    source-port 0;
    destination-port 0;
    protocol 0;
  }
  match-policy p3 {
    source 10.1.1.1/16;
    destination 239.1.1.1/32;
    source-port 0;
    destination-port 0;
    protocol 0;
  }
}
}

```

8. If you are done configuring the device, enter **commit** from configuration mode.

```

[edit]
user@host# commit

```

Configuring Member1

On member1, configure the following:

1. Configure Phase 1 SA (this configuration must match the Phase 1 SA configured on the group server).

```
[edit security group-vpn member ike]
user@member1# set proposal prop1 authentication-method pre-shared-keys dh-group
group2 authentication-algorithm sha1 encryption-algorithm 3des-cbc
user@member1# set policy pol1 mode main proposals prop1 pre-shared-key ascii-text
"$9$clgrK8-VYZUHX7UHqmF3Sre"
user@member1# set gateway g1 ike-policy pol1 address 20.0.0.1 local-address 10.1.0.1
```

2. Configure the group identifier, IKE gateway, and interface for member1.

```
[edit security group-vpn member ipsec]
user@member1# set vpn v1 group 1 ike-gateway g1 group-vpn-external-interface
ge-0/1/0
```



NOTE: To prevent packet fragmentation issues, we recommend that the interface used by the group member to connect to the MPLS network be configured for a maximum transmission unit (MTU) size no larger than 1400 bytes. Use the `set interface mtu` configuration statement to set the MTU size.

3. Configure address book entries for the 10.0.0.0/8 subnet.

```
[edit security zones]
user@member1# set security-zone trust address-book address 10_subnet 10.0.0.0/8
user@member1# set security-zone untrust address-book address 10_subnet
10.0.0.0/8
```

4. Configure a scope policy from the trust zone to the untrust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone trust to-zone untrust]
user@member1# set policy scope1 match source-address 10_subnet
destination-address 10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

5. Configure a scope policy from the untrust zone to the trust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```
[edit security policies from-zone untrust to-zone trust]
user@member1# set policy scope1 match source-address 10_subnet
destination-address 10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1
```

6. Confirm your configuration by entering the `show security group-vpn member` and `show security policies` commands from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.



NOTE: In the list of configured security policies, make sure that the scope policies are listed before the default policies.

```
[edit]
user@member1# show security group-vpn member
ike {
  proposal prop1 {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
  }
  policy pol1 {
    mode main;
    proposals prop1;
    pre-shared-key ascii-text "$9$CeS6uBEleWLNb"; ## SECRET-DATA
  }
  gateway g1 {
    ike-policy pol1;
    address 20.0.0.1;
    local-address 10.1.0.1;
  }
}
ipsec {
  vpn v1 {
    ike-gateway g1;
    group-vpn-external-interface ge-0/1/0;
    group 1;
  }
}

[edit]
user@member1# show security policies
from-zone trust to-zone trust {
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone trust to-zone untrust {
  policy scope1 {
    match {
      source-address 10_subnet;
      destination-address 10_subnet;
      application any;
    }
    then {
      permit {
```

```
        tunnel {
            ipsec-group-vpn v1;
        }
    }
}
policy default-permit {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
}
from-zone untrust to-zone trust {
    policy scope1 {
        match {
            source-address 10_subnet;
            destination-address 10_subnet;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-group-vpn v1;
                }
            }
        }
    }
    policy default-deny {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            deny;
        }
    }
}
```

7. If you are done configuring the device, enter **commit** from configuration mode.

```
[edit]
user@host# commit
```

Configuring Member2

On member2, configure the following:

1. Configure Phase 1 SA (this configuration must match the Phase 1 SA configured on the group server).

```
[edit security group-vpn member ike]
user@member2# set proposal prop2 authentication-method pre-shared-keys
dh-group group2 authentication-algorithm sha1 encryption-algorithm 3des-cbc
user@member2# set policy pol2 mode main proposals prop2 pre-shared-key ascii-text
"$9$clgrK8-VYZUHX7UHqmF3Sre"
user@member2# set gateway g2 ike-policy pol2 address 20.0.0.1 local-address
10.2.0.1
```

2. Configure the group identifier, IKE gateway, and interface for member2:

```
[edit security group-vpn member ipsec]
user@member2# set vpn v2 group 1 ike-gateway g2 group-vpn-external-interface
ge-0/1/0
```



NOTE: To prevent packet fragmentation issues, we recommend that the interface used by the group member to connect to the MPLS network be configured for a maximum transmission unit (MTU) size no larger than 1400 bytes. Use the `set interface mtu` configuration statement to set the MTU size.

3. Configure address book entries for the trust zone.

```
[edit security zones security-zone trust]
user@member2# set address-book address 10_subnet 10.0.0.0/8
user@member2# set address-book address 10_1_0_0_16 10.1.0.0/16
user@member2# set address-book address multicast_net 239.0.0.0/8
```

4. Configure address book entries for the untrust zone.

```
[edit security zones security-zone untrust]
user@member2# set address-book address 10_subnet 10.0.0.0/8
user@member2# set address-book address 10_1_0_0_16 10.1.0.0/16
user@member2# set address-book address multicast_net 239.0.0.0/8
```

5. Configure a scope policy from the trust zone to the untrust zone that blocks traffic from 10.1.0.0/16.

```
[edit security policies from-zone trust to-zone untrust]
user@member2# set policy deny2 match source-address 10_1_0_0_16
destination-address any application any
user@member2# set policy deny2 then reject
user@member2# set policy scope2 match source-address 10_subnet
destination-address 10_subnet application any
user@member2# set policy scope2 then permit tunnel ipsec-group-vpn v2
user@member2# set policy multicast-scope2 match source-address 10_subnet
destination-address multicast-net application any
user@member2# set policy multicast-scope2 then permit tunnel ipsec-group-vpn
v2
```

6. Configure a scope policy from the untrust zone to the trust zone that blocks traffic to 10.1.0.0/16.

```
[edit security policies from-zone untrust to-zone trust]
user@member2# set policy deny2 match source-address any destination-address
10_1_0_0_16 application any
user@member2# set policy deny2 then reject
user@member2# set policy scope2 match source-address 10_subnet
destination-address 10_subnet application any
```

```

user@member2# set policy scope2 then permit tunnel ipsec-group-vpn v2
user@member2# set policy multicast-scope2 match source-address 10_subnet
destination-address multicast-net application any
user@member2# set policy multicast-scope2 then permit tunnel ipsec-group-vpn
v2

```

7. Confirm your configuration by entering the **show security group-vpn member** and **show security policies** commands from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.



NOTE: In the list of configured security policies, make sure that the scope policies are listed before the default policies.

```

[edit]
user@member2# show security group-vpn member
ike {
  proposal prop2 {
    authentication-method pre-shared-keys;
    dh-group group2;
    authentication-algorithm sha1;
    encryption-algorithm 3des-cbc;
  }
  policy pol2 {
    mode main;
    proposals prop2;
    pre-shared-key ascii-text "$9$Hm5FCA0BEy"; ## SECRET-DATA
  }
  gateway g2 {
    ike-policy pol2;
    address 20.0.0.1;
    local-address 10.2.0.1;
  }
}
ipsec {
  vpn v2 {
    ike-gateway g2;
    group-vpn-external-interface ge-0/1/0;
    group 1;
  }
}

[edit]
user@member2# show security policies
from-zone trust to-zone trust {
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}

```



```
}
}
from-zone trust to-zone untrust {
  policy deny2 {
    match {
      source-address 10_1_0_0_16;
      destination-address any;
      application any;
    }
    then {
      reject;
    }
  }
  policy scope2 {
    match {
      source-address 10_subnet;
      destination-address 10_subnet;
      application any;
    }
    then {
      permit {
        tunnel {
          ipsec-group-vpn v2;
        }
      }
    }
  }
  policy multicast-scope2 {
    match {
      source-address 10_subnet;
      destination-address multicast-net;
      application any;
    }
    then {
      permit {
        tunnel {
          ipsec-group-vpn v2;
        }
      }
    }
  }
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
from-zone untrust to-zone trust {
  policy deny2 {
    match {
      source-address any;
```

```

        destination-address 10_1_0_0_16;
        application any;
    }
    then {
        reject;
    }
}
policy scope2 {
    match {
        source-address 10_subnet;
        destination-address 10_subnet;
        application any;
    }
    then {
        permit {
            tunnel {
                ipsec-group-vpn v2;
            }
        }
    }
}
policy multicast-scope2 {
    match {
        source-address 10_subnet;
        destination-address multicast-net;
        application any;
    }
    then {
        permit {
            tunnel {
                ipsec-group-vpn v2;
            }
        }
    }
}
policy default-deny {
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        deny;
    }
}
}

```

8. If you are done configuring the device, enter **commit** from configuration mode.

```

[edit]
user@host# commit

```

Viewing Dynamic Policies

After the group server downloads keys to member1, use the **show security dynamic-policies** command to view the dynamic policies installed on member1. Note that the multicast

policy p3 from the server is not installed on member1, because there is no scope policy configured on member1 that allows multicast traffic.

```
user@member1> show security dynamic-policies
Policy: scope1-0001, action-type: permit, State: enabled, Index: 1048580, AI: disabled,
Scope Policy: 4
Policy Type: Dynamic
Sequence number: 1
From zone: untrust, To zone: trust
Source addresses: 10.1.0.0/16
Destination addresses: 10.2.0.0/16
Application: Unknown
IP protocol: 0, ALG: 0, Inactivity timeout: 0
Source port range: [0-0]
Destination port range: [0-0]
Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope1-0001, action-type: permit, State: enabled, Index: 1048581, AI: disabled,
Scope Policy: 5
Policy Type: Dynamic
Sequence number: 2
From zone: trust, To zone: untrust
Source addresses: 10.1.0.0/16
Destination addresses: 10.2.0.0/16
Application: Unknown
IP protocol: 0, ALG: 0, Inactivity timeout: 0
Source port range: [0-0]
Destination port range: [0-0]
Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
```

After the group server downloads keys to member2, use the **show security dynamic-policies** command to view the dynamic policies installed on member2. Note that the policy p2 (for traffic from 10.1.0.0/16 to 10.2.0.0/16) from the server is not installed on member2, because it matches the deny2 security policy configured on member2.

```
user@member2> show security dynamic-policies
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048580, AI: disabled,
Scope Policy: 4
Policy Type: Dynamic
Sequence number: 1
From zone: untrust, To zone: trust
Source addresses: 10.1.0.0/16
Destination addresses: 10.2.0.0/16
Application: Unknown
IP protocol: 0, ALG: 0, Inactivity timeout: 0
Source port range: [0-0]
Destination port range: [0-0]
Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048580, AI: disabled,
Scope Policy: 4
Policy Type: Dynamic
Sequence number: 1
From zone: untrust, To zone: trust
Source addresses: 10.1.1.1/32
Destination addresses: 239.1.1.1/32
Application: Unknown
IP protocol: 0, ALG: 0, Inactivity timeout: 0
```

Source port range: [0-0]
Destination port range: [0-0]
Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048581, AI: disabled,
Scope Policy: 5
Policy Type: Dynamic
Sequence number: 2
From zone: trust, To zone: untrust
Source addresses: 10.2.0.0/16/0
Destination addresses: 10.1.0.0/16
Application: Unknown
IP protocol: 0, ALG: 0, Inactivity timeout: 0
Source port range: [0-0]
Destination port range: [0-0]
Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586
Policy: scope2-0001, action-type: permit, State: enabled, Index: 1048581, AI: disabled,
Scope Policy: 5
Policy Type: Dynamic
Sequence number: 2
From zone: trust, To zone: untrust
Source addresses: 10.1.1.1/32
Destination addresses: 239.1.1.1/32
Application: Unknown
IP protocol: 0, ALG: 0, Inactivity timeout: 0
Source port range: [0-0]
Destination port range: [0-0]
Tunnel: INSTANCE-gvpn_133955586, Type: IPSec, Index: 133955586

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Group VPN Overview on page 423
 - Group VPN Configuration Overview on page 431
 - Example: Configuring Group VPN with Server-Member Colocation (CLI) on page 445

Understanding Colocation Mode

Group server and group member functions are separate and do not overlap. The server and member functions can coexist in the same physical device, which is referred to as *colocation mode*. In colocation mode, there is no change in terms of functionality and behavior of the server or a member, but the server and member each need to be assigned different IP addresses so that packets can be delivered properly. In colocation mode, there can be only one IP address assigned to the server and one IP address assigned to the member across groups.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Group VPN Overview on page 423
 - Understanding Group Servers and Members on page 426
 - Understanding the GDOI Protocol on page 425
 - Understanding Dynamic Policies on page 427

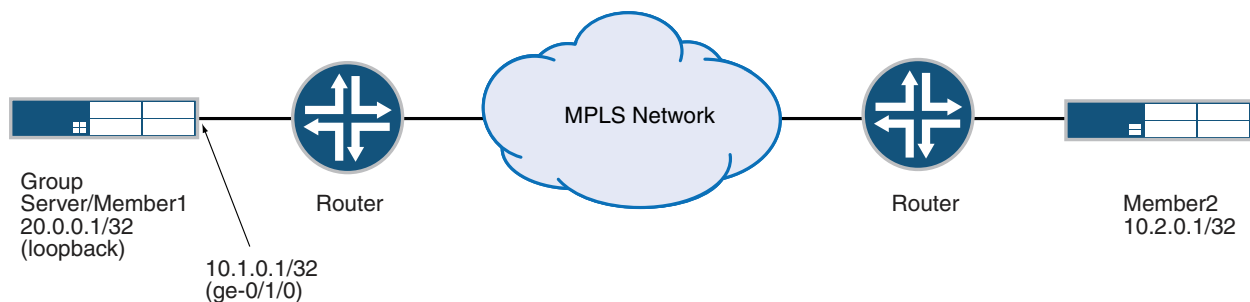
- Group VPN Configuration Overview on page 431
- Example: Configuring Group VPN (CLI) on page 432
- Example: Configuring Group VPN with Server-Member Colocation (CLI) on page 445

Example: Configuring Group VPN with Server-Member Colocation (CLI)

When colocation mode is configured, group server and group member functions can coexist in the same device. In colocation mode, the server and member must have different IP addresses so that packets are delivered properly.

This example shows you how to configure a device for colocation mode. In Figure 56 on page 445, a group VPN (group identifier is 1) consists of two members (member1 and member2) and a group server (the IP address of the loopback interface is 20.0.0.1). Note that member1 coexists in the same device as the group server. In this example, the interface that member1 uses to connect to the MPLS network (ge-0/1/0) is assigned the IP address 10.1.0.1/32.

Figure 56: Server-Member Colocation Example



NOTE: The configuration instructions in this topic describe how to configure the group server/member1 device for colocation mode. Configuration of member2 is the same as shown in the previous example.

Before you begin:

1. Configure the Juniper Networks security devices for network communication.
2. Configure network interfaces on server and member devices. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

On the group server/member1 device, configure the following:

1. Configure the loopback address on the device.

```
[edit interfaces]
user@host# set lo0 unit 0 family inet address 20.0.0.1/32
```
2. Configure the interface that member1 uses to connect to the MPLS network.

```
[edit interfaces]
```

```
user@host# set ge-0/1/0 unit 0 family inet address 10.1.0.1/32
```



NOTE: To prevent packet fragmentation issues, we recommend that the interface used by the group member to connect to the MPLS network be configured for a maximum transmission unit (MTU) size no larger than 1400 bytes. Use the `set interface mtu` configuration statement to set the MTU size.

3. Configure group VPN colocation on the device.

```
[edit security group-vpn]
user@host# set co-location
```

4. Configure IKE Phase 1 SA for the server (this configuration must match the Phase 1 SA configured on group members).

```
[edit security group-vpn server ike]
user@host# set proposal srv-prop authentication-method pre-shared-keys dh-group
group2 authentication-algorithm sha1 encryption-algorithm 3des-cbc
user@host# set policy srv-pol proposals srv-prop mode main pre-shared-key ascii-text
"$9$clgrK8-VYZUHX7UHqmF3Sre"
user@host# set gateway gw1 ike-policy srv-pol address 10.1.0.1
user@host# set gateway gw2 ike-policy srv-pol address 10.2.0.1
```

5. Configure the Phase 2 SA exchange for the server.

```
[edit security group-vpn server ipsec]
user@host# set proposal group-prop authentication-algorithm hmac-sha1-96
encryption-algorithm 3des-cbc lifetime-seconds 3600
```

6. Configure the group identifier, IKE gateway, antireplay time, and server address on the server.

```
[edit security group-vpn server group grp1]
user@host# set group-id 1 ike-gateway srv-gw anti-replay-time-window 120
server-address 20.0.0.1
```

7. Configure server to member communications.

```
[edit security group-vpn server group grp1]
user@host# set server-member-communication communication-type unicast
encryption-algorithm aes-128-cbc sig-hash-algorithm md5 certificate "srv-cert"
```

8. Configure the group policies to be downloaded to group members.

```
[edit security group-vpn server group grp1 ipsec-sa group-sa proposal group-prop]
user@host# set match-policy p1 source 10.1.0.0/16 destination 10.2.0.0/16 source-port
0 destination-port 0 protocol 0
user@host# set match-policy p2 source 10.2.0.0/16 destination 10.1.0.0/16
source-port 0 destination-port 0 protocol 0
user@host# set match-policy p3 source 10.1.1.1/16 destination 239.1.1.1/32 source-port
0 destination-port 0 protocol 0
```

9. Configure Phase 1 SA for member1 (this configuration must match the Phase 1 SA configured for the group server).

```
[edit security group-vpn member ike]
user@host# set proposal prop1 authentication-method pre-shared-keys dh-group
group2 authentication-algorithm sha1 encryption-algorithm 3des-cbc
```

```

user@host# set policy pol1 mode main proposals prop1 pre-shared-key ascii-text
"$9$c1grK8-VYZUHX7UHqmF3Sre"
user@host# set gateway g1 ike-policy pol1 address 20.0.0.1 local-address 10.1.0.1

```

10. Configure the group identifier, IKE gateway, and interface for member1.

```

[edit security group-vpn member ipsec]
user@host# set vpn v1 group1 ike-gateway g1 group-vpn-external-interface ge-0/1/0

```

11. Configure address book entries for the 10.0.0.0/8 subnet.

```

[edit security zones]
user@member1# set security-zone trust address-book address 10_subnet 10.0.0.0/8
user@member1# set security-zone untrust address-book address 10_subnet
10.0.0.0/8

```

12. Configure a scope policy from the trust zone to the untrust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```

[edit security policies from-zone trust to-zone untrust]
user@member1# set policy scope1 match source-address 10_subnet
destination-address 10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1

```

13. Configure a scope policy from the untrust zone to the trust zone that allows unicast traffic to and from the 10.0.0.0/8 subnetwork.

```

[edit security policies from-zone untrust to-zone trust]
user@member1# set policy scope1 match source-address 10_subnet
destination-address 10_subnet application any
user@member1# set policy scope1 then permit tunnel ipsec-group-vpn v1

```

14. Confirm your configuration by entering the **show security group-vpn** and **show security policies** commands from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.



NOTE: In the list of configured security policies, make sure that the scope policies are listed before the default policies.

```

[edit]
user@host# show security group-vpn
member {
  ike {
    proposal prop1 {
      authentication-method pre-shared-keys;
      dh-group group2;
      authentication-algorithm sha1;
      encryption-algorithm 3des-cbc;
    }
    policy pol1 {
      mode main;
      proposals prop1;
      pre-shared-key ascii-text "$9$i.fz9CuORS"; ## SECRET-DATA
    }
  }
  gateway g1 {
    ike-policy pol1;
  }
}

```

```
        address 20.0.0.1;
        local-address 10.1.0.1;
    }
}
ipsec {
    vpn v1 {
        ike-gateway g1;
        group-vpn-external-interface ge-0/1/0;
        group 1;
    }
}
}
server {
    ike {
        proposal srv-prop {
            authentication-method pre-shared-keys;
            dh-group group2;
            authentication-algorithm sha1;
            encryption-algorithm 3des-cbc;
        }
        policy srv-pol {
            mode main;
            proposals srv-prop;
            pre-shared-key ascii-text "$9$hucrK8-ds2aU"; ## SECRET-DATA
        }
        gateway gw1 {
            ike-policy srv-pol;
            address 10.1.0.1;
        }
        gateway gw2 {
            ike-policy srv-pol;
            address 10.2.0.1;
        }
    }
}
ipsec {
    proposal group-prop {
        authentication-algorithm hmac-sha1-96;
        encryption-algorithm 3des-cbc;
        lifetime-seconds 3600;
    }
}
group grp1 {
    group-id 1;
    ike-gateway srv-gw;
    anti-replay-time-window 120;
    server-address 20.0.0.1;
    server-member-communication {
        communication-type unicast;
        encryption-algorithm aes-128-cbc;
        sig-hash-algorithm md5;
        certificate srv-cert;
    }
    ipsec-sa group-sa {
        proposal group-prop;
        match-policy p1 {
            source 10.1.0.0/16;
        }
    }
}
```



```

        destination 10.2.0.0/16;
        source-port 0;
        destination-port 0;
        protocol 0;
    }
    match-policy p2 {
        source 10.2.0.0/16;
        destination 10.1.0.0/16;
        source-port 0;
        destination-port 0;
        protocol 0;
    }
    match-policy p3 {
        source 10.1.1.1/16;
        destination 239.1.1.1/32;
        source-port 0;
        destination-port 0;
        protocol 0;
    }
}
}
}
co-location;

[edit]
user@host# show security policies
from-zone trust to-zone trust {
    policy default-permit {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone trust to-zone untrust {
    policy scope1 {
        match {
            source-address 10_subnet;
            destination-address 10_subnet;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-group-vpn v1;
                }
            }
        }
    }
}
policy default-permit {
    match {
        source-address any;

```

```

        destination-address any;
        application any;
    }
    then {
        permit;
    }
}
}
from-zone untrust to-zone trust {
    policy default-deny {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            deny;
        }
    }
    policy scope1 {
        match {
            source-address 10_subnet;
            destination-address 10_subnet;
            application any;
        }
        then {
            permit {
                tunnel {
                    ipsec-group-vpn v1;
                }
            }
        }
    }
}
}

```

15. If you are done configuring the device, enter **commit** from configuration mode.

```

[edit]
user@host# commit

```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Group VPN Overview on page 423
 - Group VPN Configuration Overview on page 431
 - Example: Configuring Group VPN (CLI) on page 432

Understanding IKE Phase 1 Configuration for Group VPN

An IKE Phase 1 SA between the group server and a group member establishes a secure channel in which to negotiate IPsec SAs that are shared by a group. For standard IPsec VPNs on Juniper Networks security devices, Phase 1 SA configuration consists of specifying an IKE proposal, policy, and gateway. For group VPN, the IKE Phase 1 SA configuration is

similar to the configuration for standard IPsec VPNs, but is performed at the **[edit security group-vpn]** hierarchy.

In the IKE proposal configuration, you set the authentication method and the authentication and encryption algorithms that will be used to open a secure channel between participants. In the IKE policy configuration, you set the mode (main or aggressive) in which the Phase 1 channel will be negotiated, specify the type of key exchange to be used, and reference the Phase 1 proposal. In the IKE gateway configuration, you reference the Phase 1 policy.

The IKE Phase 1 configuration on the group server must match the IKE Phase 1 configuration on group members. On the server, use the **[edit security group-vpn server ike]** hierarchy to configure IKE Phase 1 SA. On a group member, use the **[edit security group-vpn member ike]** hierarchy to configure IKE Phase 1 SA.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IPsec VPN Configuration Overview on page 366
 - Group VPN Overview on page 423
 - Understanding the GDOI Protocol on page 425
 - Understanding Group Servers and Members on page 426
 - Group VPN Configuration Overview on page 431
 - Understanding IPsec SA Configuration for Group VPN on page 451

Understanding IPsec SA Configuration for Group VPN

After the server and member have established a secure and authenticated channel in Phase 1 negotiation, they proceed through Phase 2. Phase 2 negotiation establishes the IPsec SAs that are shared by group members to secure data that is transmitted among members. While the IPsec SA configuration for group VPN is similar to the configuration for standard VPNs, a group member does not need to negotiate the SA with other group members.

Phase 2 IPsec configuration for group VPN consists of the following information:

- A proposal for the security protocol, authentication, and encryption algorithm to be used for the SA. The IPsec SA proposal is configured on the group server with the **proposal** configuration statement at the **[edit security group-vpn server ipsec]** hierarchy.
- A group policy that references the proposal. A group policy specifies the traffic (protocol, source address, source port, destination address, and destination port) to which the SA and keys apply. The group policy is configured on the server with the **ipsec-sa** configuration statement at the **[edit security group-vpn server group]** hierarchy.
- An Autokey IKE that references the group identifier, the group server (configured with the **ike-gateway** configuration statement), and the interface used by the member to connect to the group. The Autokey IKE is configured on the member with the **ipsec vpn** configuration statement at the **[edit security group-vpn member]** hierarchy.



NOTE: To prevent packet fragmentation issues, we recommend that the interface used by the group member to connect to the MPLS network be configured for a maximum transmission unit (MTU) size no larger than 1400 bytes. Use the `set interface mtu` configuration statement to set the MTU size.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IPsec VPN Configuration Overview on page 366
 - Group VPN Overview on page 423
 - Understanding the GDOI Protocol on page 425
 - Understanding Group Servers and Members on page 426
 - Group VPN Configuration Overview on page 431
 - Understanding IKE Phase 1 Configuration for Group VPN on page 450

Understanding VPN Group Configuration

The VPN group is configured on the server with the **group** configuration statement at the `[edit security group-vpn server]` hierarchy.

The group information consists of the following information:

- Group identifier—A value between 1 and 65,535 that identifies the VPN group. The same group identifier must be configured on the group member for Autokey IKE.
- Group members, as configured with the **ike-gateway** configuration statement. There can be multiple instances of this configuration statement, one for each member of the group.
- IP address of the server (the loopback interface address is recommended).
- Group policies—Policies that are to be downloaded to members. Group policies describe the traffic to which the SA and keys apply. See “Understanding Dynamic Policies” on page 427.
- Server-member communication—Optional configuration that allows the server to send rekey messages to members. See “Understanding Server-Member Communication” on page 453.
- Antireplay—Optional configuration that detects packet interception and replay. See “Understanding Antireplay” on page 453.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Group VPN Overview on page 423
 - Understanding the GDOI Protocol on page 425
 - Understanding Group Servers and Members on page 426

- Group VPN Configuration Overview on page 431

Understanding Antireplay

Antireplay is an IPsec feature that can detect when a packet is intercepted and then replayed by attackers. Antireplay is enabled by default for group VPNs but can be disabled for a group with the **no-anti-replay** configuration statement.

When antireplay is enabled, the group server synchronizes the time between the group members. Each IPsec packet contains a timestamp. The group member checks whether the packet's timestamp falls within the configured **anti-replay-time-window** value (the default is 100 seconds). A packet is dropped if the timestamp exceeds the value.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - VPN Overview on page 355
 - Understanding IKE and IPsec Packet Processing on page 361
 - Understanding the GDOI Protocol on page 425
 - Understanding Group Servers and Members on page 426
 - Understanding VPN Group Configuration on page 452

Understanding Server-Member Communication

Server-member communication allows the server to send GDOI **groupkey-push** messages to members. If server-member communication is not configured for the group, members can send GDOI **groupkey-pull** messages to register and reregister with the server, but the server is not able to send rekey messages to members.

Server-member communication is configured for the group by using the **server-member-communication** configuration statement at the [edit security group-vpn server] hierarchy. The following options can be defined:

- Encryption algorithm used for communications between the server and member. You can specify 3des-cbc, aes-128-cbc, aes-192-cbc, aes-256-cbc, or des-cbc. There is no default algorithm.
- Authentication algorithm (md5 or sha1) used to authenticate the member to the server. There is no default algorithm.
- Whether the server sends unicast or multicast rekey messages to group members and parameters related to the communication type. See "Understanding Rekey Messages" on page 429.
- Interval at which the server sends heartbeat messages to the group member. This allows the member to determine whether the server has rebooted, which would require the member to reregister with the server. The default is 300 seconds. See "Understanding Heartbeat Messages" on page 457.
- Lifetime for the key encryption key (KEK). The default is 3600 seconds.



NOTE: Configuring server-member communication is necessary for the group server to send rekey messages to members, but there might be situations in which this behavior is not desired. For example, if group members are dynamic peers (such as in a home office), the devices are not always up and the IP address of a device might be different each time it is powered up. Configuring server-member communication for a group of dynamic peers can result in unnecessary transmissions by the server. If you want IKE Phase 1 SA negotiation to always be performed to protect GDOI negotiation, do not configure server-member communication.

If server-member communication for a group is not configured, the membership list displayed by the **show security group-vpn server registered-members** command shows group members who have registered with the server; members can be active or not. When server-member communication for a group is configured, the group membership list is cleared. If the communication type is configured as unicast, the **show security group-vpn server registered-members** command shows only active members. If the communication type is configured as multicast, the **show security group-vpn server registered-members** command shows members who have registered with the server after the configuration; the membership list does not necessarily represent active members because members might drop out after registration.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Group Keys on page 428
- Understanding Rekey Messages on page 429
- Understanding Member Reregistration on page 430
- Understanding VPN Group Configuration on page 452
- Example: Configuring Server-Member Communication for Unicast Rekey Messages on page 454
- Example: Configuring Server-Member Communication for Multicast Rekey Messages on page 456

Example: Configuring Server-Member Communication for Unicast Rekey Messages

This example shows the configuration that enables the server to send unicast rekey messages to group members.

Before you begin:

1. Configure the group server and members for IKE Phase 1 negotiation.
2. Configure the group server and members for Phase 2 IPsec SA.
3. On the group server, configure the group **g1**.

See “Example: Configuring Group VPN (CLI)” on page 432 or “Example: Configuring Group VPN with Server-Member Colocation (CLI)” on page 445.

Configuration instructions in this topic describe how to specify the following server-member communication for the group **g1**:

- The server sends unicast rekey messages to group members.
- 3des-cbc is used to encrypt traffic between the server and members.
- sha1 is used for member authentication.

Default values are used for server heartbeats, KEK lifetime, and retransmissions.

To configure server-member communication:

1. Set the communications type to unicast.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set communications-type unicast
```
2. Set the encryption algorithm to 3des-cbc.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set encryption-algorithm 3des-cbc
```
3. Set the member authentication to sha1.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set sig-hash-algorithm sha1
```
4. Confirm your configuration by entering the **show security group-vpn server group g1 server-member-communication** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security group-vpn server group g1 server-member-communication
communication-type unicast;
encryption-algorithm 3des-cbc;
sig-hash-algorithm sha1;
```
5. Commit the configuration if you are done configuring the device.

```
[edit]
user@host# commit
```

Related Topics • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Group VPN Configuration Overview on page 431
- Understanding Server-Member Communication on page 453
- Understanding Rekey Messages on page 429
- Understanding Group Keys on page 428
- Understanding VPN Group Configuration on page 452

Example: Configuring Server-Member Communication for Multicast Rekey Messages

This example shows the configuration that enables the server to send multicast rekey messages to group members.

Before you begin:

1. Configure the group server and members for IKE Phase 1 negotiation and Phase 2 IPsec SA. See “Example: Configuring Group VPN (CLI)” on page 432 or “Example: Configuring Group VPN with Server-Member Colocation (CLI)” on page 445.
2. On the group server, configure the group **g1**. See “Example: Configuring Group VPN (CLI)” on page 432 or “Example: Configuring Group VPN with Server-Member Colocation (CLI)” on page 445.
3. Configure the interface **ge-0/0/1.0**. This is the interface the server will use for sending multicast messages. See *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*.
4. Configure the multicast group address **226.1.1.1**. See *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*.



NOTE: IP multicast protocols must be configured to allow delivery of multicast traffic in the network. This example does not show multicast configuration. For information about configuring multicast protocols on Juniper Networks security devices, see the *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*.

Configuration instructions in this topic describe how to specify the following server-member communication for the group **g1**:

- The server sends multicast rekey messages to group members by means of the multicast address **226.1.1.1** and interface **ge-0/0/1.0**.
- **3des-cbc** is used to encrypt traffic between the server and members.
- **sha1** is used for member authentication.

Default values are used for server heartbeats, KEK lifetime, and retransmissions.

To configure server-member communication:

1. Set the communications type to multicast.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set communications-type multicast
```
2. Set the multicast group to **226.1.1.1**.

```
[edit security group-vpn server group g1 server-member-communication]
user@host# set multicast-group 226.1.1.1
```
3. Set the **ge-0/0/1.0** interface for outgoing multicast messages.

```
[edit security group-vpn server group g1 server-member-communication]
```



```
user@host# set multicast-outgoing-interface ge-0/0/1.0
```

4. Set the encryption algorithm to 3des-cbc.

```
[edit security group-vpn server group g1 server-member-communication]
```

```
user@host# set encryption-algorithm 3des-cbc
```

5. Set the member authentication to sha1.

```
[edit security group-vpn server group g1 server-member-communication]
```

```
user@host# set sig-hash-algorithm sha1
```

6. Confirm your configuration by entering the **show security group-vpn server group g1 server-member-communication** command from configuration mode. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
```

```
user@host# show security group-vpn server group g1 server-member-communication
communication-type multicast;
multicast-group 226.1.1.1;
multicast-outgoing-interface ge-0/0/1.0;
encryption-algorithm 3des-cbc;
sig-hash-algorithm sha1;
```

7. Commit the configuration if you are done configuring the device.

```
[edit]
```

```
user@host# commit
```

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Group VPN Configuration Overview on page 431
- Understanding Server-Member Communication on page 453
- Understanding Rekey Messages on page 429
- Understanding Group Keys on page 428
- Understanding VPN Group Configuration on page 452

Understanding Heartbeat Messages

When server-member communication is configured, the server sends heartbeat messages to members at specified intervals (the default interval is 300 seconds). The heartbeat mechanism allows members to reregister with the server if the specified number of heartbeats is not received. For example, members will not receive heartbeat messages during a server reboot. When the server has rebooted, members reregister with the server.

Heartbeats are transmitted through **groupkey-push** messages. The sequence number is incremented on each heartbeat message, which protects members from reply attacks. Unlike rekey messages, heartbeat messages are not acknowledged by recipients and are not retransmitted by the server.

Heartbeat messages contain the following information:

- Current state and configuration of the keys on the server
- Relative time, if antireplay is enabled

By comparing the information in the heartbeats, a member can detect whether it has missed server information or rekey messages. The member reregisters to synchronize itself with the server.



NOTE: Heartbeat messages can increase network congestion and cause unnecessary member reregistrations. Thus, heartbeat detection can be disabled on the member if necessary.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Group VPN Configuration Overview on page 431
- Understanding Server-Member Communication on page 453
- Understanding the GDOI Protocol on page 425
- Understanding Group Servers and Members on page 426

Understanding Group VPN Limitations

The following are not supported f in this release for group VPNs:

- Non-default routing instances
- Chassis cluster
- Server clusters
- Route-based group VPN
- Public Internet-based deployment
- SNMP
- Deny policy from Cisco GET VPN server
- J-Web interface for configuration and monitoring

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Group VPN Overview on page 423
- Understanding the GDOI Protocol on page 425
- Understanding Group Servers and Members on page 426

Understanding Interoperability with Cisco GET VPN

Cisco's implementation of GDOI is called Group Encryption Transport (GET) VPN. While group VPN in Junos OS and Cisco's GET VPN are both based on RFC 3547, *The Group*

Domain of Interpretation, there are some implementation differences that you need to be aware of when deploying GDOI in a networking environment that includes both Juniper Networks security devices and Cisco routers. For more information, see the current Junos OS Release notes.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - VPN Overview on page 355
 - Understanding IKE and IPsec Packet Processing on page 361
 - Understanding the GDOI Protocol on page 425
 - Understanding Group Servers and Members on page 426

PART 7

Intrusion Detection and Prevention

- IDP Policies on page 463
- Application-Level Distributed Denial of Service on page 523
- IDP Signature Database on page 535
- IDP Application Identification on page 549
- IDP SSL Inspection on page 563
- IDP Performance and Capacity Tuning on page 569
- IDP Logging on page 571

CHAPTER 22

IDP Policies

- IDP Policies Overview on page 463
- Example: Enabling IDP in a Security Policy on page 465
- IDP Inline Tap Mode on page 468
- IDP Rules and Rulebases on page 470
- IDP Applications and Application Sets on page 493
- IDP Attacks and Attack Objects on page 496
- Limitations of IDP on page 519

IDP Policies Overview

The Junos OS Intrusion Detection and Prevention (IDP) policy enables you to selectively enforce various attack detection and prevention techniques on network traffic passing through an IDP-enabled device. It allows you to define policy rules to match a section of traffic based on a zone, network, and application, and then take active or passive preventive actions on that traffic.

An IDP policy defines how your device handles the network traffic. It allows you to enforce various attack detection and prevention techniques on traffic traversing your network.

A policy is made up of *rulebases* and each rulebase contains a set of *rules*. You define rule parameters, such as traffic match conditions, action, and logging requirements, then add the rules to rule bases. After you create an IDP Policy by adding rules in one or more rulebases, you can select that policy to be the active policy on your device.

Junos OS allows you to configure multiple IDP policies, but a device can have only one active IDP policy at a time. You can install the same IDP policy on multiple devices, or you can install a unique IDP policy on each device in your network. A single policy can contain only one instance of any type of rulebase.

This topic includes the following sections:

- IDP Policy Terms on page 464
- Working with IDP Policies on page 464

IDP Policy Terms

Before configuring IDP policies, become familiar with the terms defined in Table 41 on page 464.

Table 41: IDP Terms

Term	Definition
Attacks	Attacks attempt to exploit vulnerabilities in computer hardware and software. Depending on the severity of the attack, it might disable your system completely, allow an attacker to gain confidential information stored on your system, or use your network to attack other networks.
Attack objects	A signature or protocol anomaly that is combined with context information. Attack objects are used in Main rulebase rules to match malicious traffic patterns. Each attack object detects a known attack or protocol anomaly that can be used by an attacker to compromise your network.
False positives	Any situation in which benign traffic causes an intrusion detection service to generate an alert; also known as a false alert.
Protocol anomaly	A deviation from the RFC specifications that dictate how communications between two entities should be implemented. Most legitimate traffic does not deviate from the protocols; when anomalies are detected, they are often a sign of malicious traffic and seen as a threat to the system.
Rule	A user-defined match/action sequence. Rules are represented graphically in the Security Policy Editor, where you can create, modify, delete, and reorder them in a rulebase.
Rulebase	A set of rules that uses a specific detection mechanism to identify and prevent attacks.
Severity	The designated threat level of an attack (critical, high, medium, low, or informational). Attack objects use the severity setting that matches the threat level of the attack they detect.

Working with IDP Policies

You can perform the following tasks to manage IDP policies:

- Create new IDP policies starting from scratch. See “Example: Defining Rules for an IDP IPS Rulebase” on page 481.
- Create an IDP policy starting with one of the predefined templates provided by Juniper Networks (see “Understanding Predefined IDP Policy Templates” on page 537).
- Add or delete rules within a rulebase. You can use any of the following IDP objects to create rules:
 - Zone and network objects available in the base system
 - Predefined service objects provided by Juniper Networks

- Custom application objects
- Predefined attack objects provided by Juniper Networks
- Create custom attack objects (see “Example: Configuring IDP Signature-Based Attacks” on page 515).
- Update the signature database provided by Juniper Networks. This database contains all predefined objects.
- Maintain multiple IDP policies. Any one of the policies can be applied to the device.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Policy Rules on page 470
 - Understanding IDP Terminal Rules on page 487
 - Understanding IDP Application Sets on page 493
 - Understanding Custom Attack Objects on page 496
 - Understanding Predefined IDP Policy Templates on page 537
 - Example: Enabling IDP in a Security Policy on page 465

Example: Enabling IDP in a Security Policy

This example shows how to configure two security policies to enable IDP services on all traffic flowing in both directions on the device.

- Requirements on page 465
- Overview on page 465
- Configuration on page 466
- Verification on page 468

Requirements

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Create security zones. See “Example: Creating Security Zones” on page 88.
- Configure applications. See “Example: Configuring IDP Applications and Services” on page 494.

Overview

For transit traffic to pass through IDP inspection, you configure a security policy and enable IDP application services on all traffic that you want to inspect. Security policies contain rules defining the types of traffic permitted on the network and the way that the

traffic is treated inside the network. Enabling IDP in a security policy directs traffic that matches the specified criteria to be checked against the IDP rulebases.

To allow transit traffic to pass through without IDP inspection, specify a *permit* action for the rule without enabling the IDP application services. Traffic matching the conditions in this rule passes through the device without IDP inspection.

This example shows how to configure two policies, **idp-app-policy-1** and **idp-app-policy-2**, to enable IDP services on all traffic flowing in both directions on the device. Policy **idp-app-policy-1** directs all traffic flowing from previously configured zones **Zone1** to **Zone2** to be checked against IDP rulebases. The policy **idp-app-policy-2** directs all traffic flowing from **Zone2** to **Zone1** to be checked against IDP rulebases.



NOTE: The action set in the security policy action must be *permit*. You cannot enable IDP for traffic that the device denies or rejects.

Configuration

CLI Quick Configuration

To quickly configure two policies, **idp-app-policy-1** and **idp-app-policy-2**, to enable IDP services on all traffic flowing in both directions on the device, copy the following commands and paste them into the CLI:

```
[edit]
set security policies from-zone Zone1 to-zone Zone2 policy idp-app-policy-1
set security policies from-zone Zone1 to-zone Zone2 policy idp-app-policy-1 match
  source-address any destination-address any application any
set security policies from-zone Zone1 to-zone Zone2 policy idp-app-policy-1 then permit
  application-services idp
set security policies from-zone Zone2 to-zone Zone1 policy idp-app-policy-2
set security policies from-zone Zone2 to-zone Zone1 policy idp-app-policy-2 match
  source-address any destination-address any application any
set security policies from-zone Zone2 to-zone Zone1 policy idp-app-policy-2 then permit
  application-services idp
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure two policies, **idp-app-policy-1** and **idp-app-policy-2**, to enable IDP services on all traffic flowing in both directions on the device:

1. Create a security policy for traffic traversing from **Zone1** to **Zone2**.

```
[edit]
user@host# set security policies from-zone Zone1 to-zone Zone2 policy
  idp-app-policy-1
```

2. Specify the match conditions for the traffic flowing in one direction

```
[edit security policies from-zone Zone1 to-zone Zone2 policy idp-app-policy-1]
user@host# set match source-address any destination-address any application
  any
```

3. Specify the action to be taken on traffic that matches the specified conditions.

```
[edit security policies from-zone Zone1 to-zone Zone2 policy idp-app-policy-1]
user@host# set then permit application-services idp
```
4. Create another security policy for traffic traversing in the other direction from **Zone2** to **Zone1**.

```
[edit]
user@host# set security policies from-zone Zone2 to-zone Zone1 policy
idp-app-policy-2
```
5. Specify the match conditions for the traffic flowing in the other direction.

```
[edit security policies from-zone Zone2 to-zone Zone1 policy idp-app-policy-2]
user@host# set match source-address any destination-address any application
any
```
6. Specify the action to be taken on traffic that matches the conditions specified in the policy.

```
[edit security policies from-zone Zone2 to-zone Zone1 policy idp-app-policy-2]
user@host# set then permit application-services idp
```

Results From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security policies
from-zone Zone-1 to-zone Zone-2 {
  policy idp-app-policy-1 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit {
        application-services {
          idp;
        }
      }
    }
  }
}
from-zone Zone-2 to-zone Zone-1 {
  policy idp-app-policy-2 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit {
        application-services {
          idp;
        }
      }
    }
  }
}
```

```
    }  
  }  
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration on page 468

Verifying the Configuration

Purpose Verify if the security policy configuration is correct.

Action From operational mode, enter the **show security policies** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP Policies Overview on page 463
 - Understanding IDP Policy Rules on page 470
 - Understanding IDP Policy Rulebases on page 476

IDP Inline Tap Mode

- Understanding IDP Inline Tap Mode on page 468
- Example: Configuring IDP Inline Tap Mode on page 469

Understanding IDP Inline Tap Mode

The main purpose of inline tap mode is to provide best case deep inspection analysis of traffic while maintaining over all performance and stability of the device. The inline tap feature provides passive, inline detection of application layer threats for traffic matching security policies which have the IDP application service enabled. When a device is in inline tap mode, packets pass through firewall inspection and are also copied to the independent IDP module. This allows the packets to get to the next service module without waiting for IDP processing results. By doing this, when the traffic input is beyond the IDP throughput limit, the device can still sustain processing as long as it does not go beyond the modules limits, such as with the firewall. If the IDP process fails, all other features of the device will continue to function normally. Once the IDP process recovers, it will resume processing packets for inspection. Since inline tap mode puts IDP in a passive mode for monitoring, preventative actions such as session close, drop, and mark diffserv are deferred. The action drop packet is ignored.

Inline tap mode can only be configured if the forwarding process mode is set to maximize IDP sessions, which ensures stability and resiliency for firewall services. You also do not need a separate tap or span port to use inline tap mode.



NOTE: You must restart the device when switching to inline tap mode or back to regular mode.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Configuring IDP Inline Tap Mode on page 469
- IDP Policies Overview on page 463
- Understanding IDP Policy Rules on page 470
- Understanding IDP Policy Rulebases on page 476

Example: Configuring IDP Inline Tap Mode

This example shows how to configure a device for inline tap mode.

Requirements

Before you begin, review the inline tap mode feature. See “Understanding IDP Inline Tap Mode” on page 468.

Overview

The inline tap mode feature provides passive, inline detection of application layer threats for traffic matching security policies which have the IDP application service enabled.



NOTE: IDP inline tap mode does not require a separate tap or span port.

Configuration

Step-by-Step Procedure

To configure a device for inline tap mode:

1. Set inline tap mode.

```
[edit security]
user@host# set forwarding-process application-services maximize-idp-sessions
inline-tap
```
2. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```
3. Restart the system from operational mode.

```
user@host> request system reboot
```



NOTE: When switching to inline tap mode or back to regular mode, you must restart the device .

4. If you want to switch the device back to regular mode, delete the inline tap mode.

```
[edit security]
user@host# delete forwarding-process application-services maximize-idp-sessions
inline-tap
```

Verification

To verify that inline tap mode is enabled, enter the **show security idp status** command. The forwarding process mode line item will show “Forwarding process mode : maximizing sessions (Inline-tap)”

- Related Topics**
- IDP Policies Overview on page 463
 - Understanding IDP Policy Rules on page 470
 - Understanding IDP Policy Rulebases on page 476

IDP Rules and Rulebases

- Understanding IDP Policy Rules on page 470
- IDP Rulebases on page 476
- Understanding IDP Application-Level DDoS Rulebases on page 479
- IDP IPS Rulebase on page 480
- IDP Exempt Rulebase on page 484
- IDP Terminal Rules on page 487
- IDP DSCP Rules on page 490

Understanding IDP Policy Rules

Each instruction in an Intrusion Detection and Prevention (IDP) policy is called a rule. Rules are created in rulebases.

Rulebases are a set of rules that combine to define an IDP policy. Rules provide context to detection mechanisms by specifying which part of the network traffic the IDP system should look in to find attacks. When a rule is matched, it means that an attack has been detected in the network traffic, triggering the action for that rule. The IDP system performs the specified action and protects your network from that attack.

IDP policy rules are made up of the following components:

- Understanding IDP Rule Match Conditions on page 471
- Understanding IDP Rule Objects on page 471
- Understanding IDP Rule Actions on page 473
- Understanding IDP Rule IP Actions on page 475
- Understanding IDP Rule Notifications on page 475

Understanding IDP Rule Match Conditions

Match conditions specify the type of network traffic you want IDP to monitor for attacks.

Match conditions use the following characteristics to specify the type of network traffic to be monitored:

- **From-zone and to-zone**—All traffic flows from a source to a destination zone. You can select any zone for the source or destination. You can also use zone exceptions to specify unique to and from zones for each device. Specify **any** to monitor network traffic originating from and to any zone. The default value is **any**.
- **Source IP Address**—Specify the source IP address from which the network traffic originates. You can specify **any** to monitor network traffic originating from any IP address. You can also specify **source-except** to specify all sources except the specified addresses. The default value is **any**.
- **Destination IP address**—Specify the destination IP address to which the network traffic is sent. You can set this to **any** to monitor network traffic sent to any IP address. You can also specify **destination-except** to specify all destinations except the specified addresses. The default value is **any**.
- **Application**—Specify the Application Layer protocols supported by the destination IP address. You can specify **any** for all applications and **default** for the application configured in the attack object for the rule.

Understanding IDP Rule Objects

Objects are reusable logical entities that you can apply to rules. Each object that you create is added to a database for the object type.

You can configure the following types of objects for IDP rules.

Zone Objects

A zone or security zone is a collection of one or more network interfaces. IDP uses zone objects configured in the base system.

Address or Network Objects

Address objects represent components of your network, such as host machines, servers, and subnets. You use address objects in IDP policy rules to specify the network components that you want to protect.

Application or Service Objects

Service objects represent network services that use Transport Layer protocols such as TCP, UDP, RPC, and ICMP. You use service objects in rules to specify the service an attack uses to access your network. Juniper Networks provides predefined service objects, a database of service objects that are based on industry-standard services. If you need to add service objects that are not included in the predefined service objects, you can create custom service objects. IDP supports the following types of service objects:

- **Any**—Allows IDP to match all Transport Layer protocols.
- **TCP**—Specifies a TCP port or a port range to match network services for specified TCP ports. You can specify **junos-tcp-any** to match services for all TCP ports.
- **UDP**—Specifies a UDP port or a port range to match network services for specified UDP ports. You can specify **junos-udp-any** to match services for all UDP ports.
- **RPC**—Specifies a remote procedure call (RPC from Sun Microsystems) program number or a program number range. IDP uses this information to identify RPC sessions.
- **ICMP**—Specifies a type and code that is a part of an ICMP packet. You can specify **junos-icmp-all** to match all ICMP services.

Attack Objects

IDP attack objects represent known and unknown attacks. IDP includes a predefined attack object database that is periodically updated by Juniper Networks. Attack objects are specified in rules to identify malicious activity. Each attack is defined as an attack object, which represents a known pattern of attack. Whenever this known pattern of attack is encountered in the monitored network traffic, the attack object is matched. The three main types of attack objects are described in Table 42 on page 472:

Table 42: IDP Attack Objects Description

Attack Objects	Description
Signature Attack Objects	Signature attack objects detect known attacks using stateful attack signatures. An attack signature is a pattern that always exists within an attack; if the attack is present, so is the attack signature. With stateful signatures, IDP can look for the specific protocol or service used to perpetrate the attack, the direction and flow of the attack, and the context in which the attack occurs. Stateful signatures produce few false positives because the context of the attack is defined, eliminating huge sections of network traffic in which the attack would not occur.
Protocol Anomaly Attack Objects	Protocol anomaly attack objects identify unusual activity on the network. They detect abnormal or ambiguous messages within a connection according to the set of rules for the particular protocol being used. Protocol anomaly detection works by finding deviations from protocol standards, most often defined by RFCs and common RFC extensions. Most legitimate traffic adheres to established protocols. Traffic that does not, produces an anomaly, which may be created by attackers for specific purposes, such as evading an intrusion prevention system (IPS).

Table 42: IDP Attack Objects Description (*continued*)

Attack Objects	Description
Compound Attack Objects	A compound attack object combines multiple signatures and/or protocol anomalies into a single object. Traffic must match all of the combined signatures and/or protocol anomalies to match the compound attack object; you can specify the order in which signatures or anomalies must match. Use compound attack objects to refine your IDP policy rules, reduce false positives, and increase detection accuracy. A compound attack object enables you to be very specific about the events that need to occur before IDP identifies traffic as an attack. You can use And , Or , and Ordered and operations to define the relationship among different attack objects within a compound attack and the order in which events occur.

Attack Object Groups

IDP contains a large number of predefined attack objects. To help keep IDP policies organized and manageable, attack objects can be grouped. An attack object group can contain one or more attack objects of different types. Junos OS supports the following two types of attack groups:

- Static groups—Contain a fixed set of attack objects.
- Dynamic groups—Contain attack objects based on a certain matching criteria. For example, a dynamic group can contain all attacks related to an application. During signature update, the dynamic group membership is automatically updated based on the matching criteria for that group.

Understanding IDP Rule Actions

Actions specify the actions you want IDP to take when the monitored traffic matches the attack objects specified in the rules.

Table 43 on page 473 shows the actions you can specify for IDP rules:

Table 43: IDP Rule Actions

Term	Definition
No Action	No action is taken. Use this action when you only want to generate logs for some traffic.
Ignore Connection	Stops scanning traffic for the rest of the connection if an attack match is found. IDP disables the rulebase for the specific connection. NOTE: This action does not mean ignore an attack.
Diffserv Marking	Assigns the indicated Differentiated Services code point (DSCP) value to the packet in an attack, then passes the packet on normally. Note that DSCP value is not applied to the first packet that is detected as an attack, but is applied to subsequent packets.

Table 43: IDP Rule Actions (*continued*)

Term	Definition
Drop Packet	<p>Drops a matching packet before it can reach its destination but does not close the connection. Use this action to drop packets for attacks in traffic that is prone to spoofing, such as UDP traffic. Dropping a connection for such traffic could result in a denial of service that prevents you from receiving traffic from a legitimate source-IP address.</p> <p>NOTE: When an IDP policy is configured using a non-packet context defined in a custom signature for any application and has the action drop packet, when IDP identifies an attack the decoder will promote drop_packet to drop_connection. With a DNS protocol attack, this is not the case. The DNS decoder will not promote drop_packet to drop_connection when an attack is identified. This will ensure that only DNS attack traffic will be dropped and valid DNS requests will continue to be processed. This will also avoid TCP retransmission for the valid TCP DNS requests..</p>
Drop Connection	Drops all packets associated with the connection, preventing traffic for the connection from reaching its destination. Use this action to drop connections for traffic that is not prone to spoofing.
Close Client	Closes the connection and sends an RST packet to the client but not to the server.
Close Server	Closes the connection and sends an RST packet to the server but not to the client.
Close Client and Server	Closes the connection and sends an RST packet to both the client and the server.
Recommended	<p>All predefined attack objects have a default action associated with them. This is the action that Juniper Networks recommends when that attack is detected.</p> <p>NOTE: This action is supported only for IPS rulebases.</p> <p>Recommended —A list of all attack objects that Juniper Networks considers to be serious threats, organized into categories.</p> <ul style="list-style-type: none"> • Attack type groups attack objects by type (anomaly or signature). Within each type, attack objects are grouped by severity. • Category groups attack objects by predefined categories. Within each category, attack objects are grouped by severity. • Operating system groups attack objects by the operating system to which they apply: BSD, Linux, Solaris, or Windows. Within each operating system, attack objects are grouped by services and severity. • Severity groups attack objects by the severity assigned to the attack. IDP has five severity levels: Critical, Major, Minor, Warning, and Info. Within each severity, attack objects are grouped by category.

Understanding IDP Rule IP Actions

IP actions are actions that apply on future connections that use the same IP action attributes. For example, you can configure an IP action in the rule to block all future HTTP sessions between two hosts if an attack is detected on a session between the hosts. Or you can specify a timeout value that defines that the action should be applied only if new sessions are initiated within that specified timeout value. The default timeout value for IP actions is 0, which means that IP actions are never timed out.

IP actions are similar to other actions; they direct IDP to drop or close the connection. However, because you now also have the attacker's IP address, you can choose to block the attacker for a specified time. If attackers cannot immediately regain a connection to your network, they might try to attack easier targets. Use IP actions in conjunction with actions and logging to secure your network.

IP action attributes are a combination of the following fields:

- Source IP address
- Destination IP address
- Destination port
- From-zone
- Protocol

Table 44 on page 475 summarizes the types IP actions supported by IDP rules:

Table 44: IDP Rule IP Actions

Term	Definition
Notify	Does not take any action against future traffic, but logs the event. This is the default.
Drop/Block Session	All packets of any session matching the IP action rule are dropped silently.
Close Session	Any new sessions matching this IP action rule are closed by sending RST packets to the client and server.

When traffic matches multiple rules, the most severe IP action of all matched rules is applied. The most severe IP action is the Drop/Block Session action, the next in severity is the Close Session action, and then the Notify action.

Understanding IDP Rule Notifications

Notification defines how information is to be logged when an action is performed. When attacks are detected, you can choose to log an attack and create log records with attack information and send that information to the log server.

By using notifications, you can also configure the following options that instruct the log server to perform specific actions on logs generated for each rule:

- **Set Alerts**—Specify an alert option for a rule in the IDP policy. When the rule is matched, the corresponding log record displays an alert in the alert column of the Log Viewer. Security administrators use alerts to become aware of and react to important security events.
- **Set Severity Level**—Set severity levels in logging to support better organization and presentation of log records on the log server. You can use the default severity settings of the selected attack objects or choose a specific severity for your rule. The severity you configure in the rules overrides the inherited attack severity. You can set the severity level to the following levels:
 - Info—2
 - Warning—3
 - Minor—4
 - Major—5
 - Critical—7

Related Topics • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Understanding IDP Policy Rulebases on page 476
- Understanding IDP Application-Level DDoS Rulebases on page 479
- Understanding IDP IPS Rulebases on page 480
- Understanding IDP Exempt Rulebases on page 484
- Understanding IDP Terminal Rules on page 487
- Understanding DSCP Rules in IDP Policies on page 490
- Understanding Predefined IDP Policy Templates on page 537

IDP Rulebases

- Understanding IDP Policy Rulebases on page 476
- Example: Inserting a Rule in the IDP Rulebase on page 477
- Example: Deactivating and Reactivating Rules in a IDP Rulebase on page 478

Understanding IDP Policy Rulebases

Intrusion Detection and Prevention (IDP) policies are collections of rules and rulebases. A rulebase is an ordered set of rules that use a specific detection method to identify and prevent attacks.

Rules are instructions that provide context to detection mechanisms by specifying which part of the network traffic the IDP system should look in to find attacks. When a rule is matched, it means that an attack has been detected in the network traffic, triggering the action for that rule. The IDP system performs the specified action and protects your network from that attack.

Each rulebase can have multiple rules—you determine the sequence in which rules are applied to network traffic by placing them in the desired order. Each rulebase in the IDP system uses specific detection methods to identify and prevent attacks. Junos OS supports two types of rulebases—intrusion prevention system (IPS) rulebase and exempt rulebase.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Policy Rules on page 470
 - Understanding IDP Application-Level DDoS Rulebases on page 479
 - Understanding IDP IPS Rulebases on page 480
 - Understanding IDP Exempt Rulebases on page 484
 - Example: Inserting a Rule in the IDP Rulebase on page 477
 - Example: Deactivating and Activating Rules in an IDP Rulebase on page 478

Example: Inserting a Rule in the IDP Rulebase

This example shows how to insert a rule in the rulebase.

Requirements

Before you begin:

- Configure network interfaces. See *Junos OS Interfaces Configuration Guide for Security Devices*.
- Define rules in a rulebase. See “Example: Defining Rules for an IDP IPS Rulebase” on page 481.

Overview

The IDP rule-matching algorithm starts from the top of the rulebase and checks traffic against all rules in the rulebase that match the specified match conditions. You determine the sequence in which rules are applied to network traffic by placing them in the desired order. When you add a rule to the rulebase, it is placed at the end of the existing list of rules. To place a rule in any other location than at the end of the rulebase, you *insert* the rule at the desired location in the rulebase. This example places rule **R2** before rule **R1** in the IPS rulebase in a policy called **base-policy**.

Configuration

Step-by-Step Procedure

To insert a rule in the rulebase:

1. Define the position of the rule in the rulebase based on the order in which you want the rule to be evaluated.


```
[edit]
user@host# insert security idp idp-policy base-policy rulebase-ips rule R2 before rule R1
```
2. If you are done configuring the device, commit the configuration.


```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security idp** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding IDP Policy Rules on page 470
- Understanding IDP Policy Rulebases on page 476
- Example: Defining Rules for an IDP Exempt Rulebase on page 485
- Example: Enabling IDP Protection Against Application-Level DDoS Attacks (CLI) on page 528
- Example: Enabling IDP in a Security Policy on page 465

Example: Deactivating and Reactivating Rules in a IDP Rulebase

This example shows how to deactivate and activate a rule in a rulebase.

Requirements

Before you begin:

- Configure network interfaces. See *Junos OS Interfaces Configuration Guide for Security Devices*.
- Define rules in a rulebase. See “Example: Defining Rules for an IDP IPS Rulebase” on page 481.

Overview

In a rulebase, you can disable and enable rules by using the **deactivate** and **activate** commands. The **deactivate** command comments out the specified statement from the configuration. Rules that have been deactivated do not take effect when you issue the **commit** command. The **activate** command adds the specified statement back to the configuration. Rules that have been activated take effect when you next issue the **commit** command. This example shows how to deactivate and reactivate rule **R2** in an IPS rulebase that is associated with a policy called **base-policy**.

Configuration

Step-by-Step Procedure

To deactivate and activate a rule in a rulebase:

1. Specify the rule that you want to deactivate.

```
[edit]  
user@host# deactivate security idp idp-policy base-policy rulebase-ips rule R2
```
2. To reactivate the rule, use the **activate** command.

```
[edit]  
user@host# activate security idp idp-policy base-policy rulebase-ips rule R2
```
3. If you are done configuring the device, commit the configuration.

```
[edit]  
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security idp** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding IDP Policy Rules on page 470
- Understanding IDP Policy Rulebases on page 476
- Example: Defining Rules for an IDP Exempt Rulebase on page 485
- Example: Enabling IDP Protection Against Application-Level DDoS Attacks (CLI) on page 528
- Example: Enabling IDP in a Security Policy on page 465

Understanding IDP Application-Level DDoS Rulebases

The application-level DDoS rulebase defines parameters used to protect servers, such as DNS or HTTP, from application-level distributed denial-of-service (DDoS) attacks. You can set up custom application metrics based on normal server activity requests to determine when clients should be considered an attack client. The application-level DDoS rulebase is then used to define the source match condition for traffic that should be monitored, then takes the defined action: close server, drop connection, drop packet, or no action. It can also perform an IP action: ip-block, ip-close, ip-notify, or timeout. Table 45 on page 479 summarizes the options that you can configure in the application-level DDoS rulebase rules.

Table 45: Application-Level DDoS Rulebase Components

Term	Definition
Match condition	Specify the network traffic you want the device to monitor for attacks.
Action	Specify the actions you want Intrusion Detection and Prevention (IDP) to take when the monitored traffic matches the application-ddos objects specified in the application-level DDoS rule.
IP Action	Enables you to implicitly block a source address to protect the network from future intrusions while permitting legitimate traffic. You can configure one of the following IP action options in application-level DDoS: ip-block, ip-close, and ip-notify.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP Policies Overview on page 463
- Understanding IDP Policy Rulebases on page 476
- Understanding IDP Policy Rules on page 470
- IDP Application-Level DDoS Attack Overview on page 523
- IDP Application-Level DDoS Protection Overview on page 523

- Example: Enabling IDP Protection Against Application-Level DDoS Attacks (CLI) on page 528

IDP IPS Rulebase

- Understanding IDP IPS Rulebases on page 480
- Example: Defining Rules for an IDP IPS Rulebase on page 481

Understanding IDP IPS Rulebases

The intrusion prevention system (IPS) rulebase protects your network from attacks by using attack objects to detect known and unknown attacks. It detects attacks based on stateful signature and protocol anomalies. Table 46 on page 480 summarizes the options that you can configure in the IPS-rulebase rules.

Table 46: IPS Rulebase Components

Term	Definition
Match condition	Specify the type of network traffic you want the device to monitor for attacks. For more information about match conditions, see “Understanding IDP Rule Match Conditions” on page 471.
Attack objects/groups	Specify the attacks you want the device to match in the monitored network traffic. Each attack is defined as an attack object, which represents a known pattern of attack. For more information about attack objects, see “Understanding IDP Rule Objects” on page 471.
Terminal flag	Specify a terminal rule. The device stops matching rules for a session when a terminal rule is matched. For more information about terminal rules, see “Understanding IDP Terminal Rules” on page 487.
Action	Specify the action you want the system to take when the monitored traffic matches the attack objects specified in the rules. If an attack triggers multiple rule actions, then the most severe action among those rules is executed. For more information about actions, see “Understanding IDP Policy Rules” on page 470.
IP Action	Enables you to protect the network from future intrusions while permitting legitimate traffic. You can configure one of the following IP action options in the IPS rulebase—notify, drop, or close. For more information about IP actions, see “Understanding IDP Policy Rules” on page 470.
Notification	Defines how information is to be logged when action is performed. You can choose to log an attack, create log records with the attack information, and send information to the log server. For more information, see “Understanding IDP Policy Rules” on page 470.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Policy Rules on page 470
 - Understanding IDP Policy Rulebases on page 476

- Understanding IDP Exempt Rulebases on page 484
- Understanding IDP Terminal Rules on page 487
- Understanding Predefined IDP Policy Templates on page 537
- Example: Defining Rules for an IDP IPS Rulebase on page 481

Example: Defining Rules for an IDP IPS Rulebase

This example shows how to define rules for an IDP IPS rulebase.

- Requirements on page 481
- Overview on page 481
- Configuration on page 482
- Verification on page 483

Requirements

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Create security zones. See “Example: Creating Security Zones” on page 88.
- Enable IDP in security policies. See “Example: Enabling IDP in a Security Policy” on page 465.

Overview

Each rule is composed of match conditions, objects, actions, and notifications. When you define an IDP rule, you must specify the type of network traffic you want IDP to monitor for attacks by using the following characteristics—source zone, destination zone, source IP address, destination IP address, and the Application Layer protocol supported by the destination IP address. The rules are defined in rulebases, and rulebases are associated with policies.

This example describes how to create a policy called **base-policy**, specify a rulebase for this policy, and then add a rule **R1** to this rulebase. In this example, rule **R1**:

- Specifies the match condition to include any traffic from a previously configured zone called *trust* to another previously configured zone called *untrust*. The match condition also includes a predefined attack group **Critical - TELNET**. The application setting in the match condition is *default* and matches any application configured in the attack object.
- Specifies an action to drop connection for any traffic that matches the criteria for rule **R1**,
- Enables attack logging and specifies that an alert flag is added to the attack log.
- Specifies a severity level as *critical*.

After defining the rule, you specify **base-policy** as the active policy on the device.

Configuration

CLI Quick Configuration To quickly define rules for an IDP IPS rulebase, copy the following commands and paste them into the CLI.

```
[edit]
set security idp idp-policy base-policy
set security idp idp-policy base-policy rulebase-ips
set security idp idp-policy base-policy rulebase-ips rule R1
set security idp idp-policy base-policy rulebase-ips rule R1 match from-zone trust to-zone
  untrust source-address any destination-address any application default
set security idp idp-policy base-policy rulebase-ips rule R1 match attacks
  predefined-attack-groups "Critical-TELNET"
set security idp idp-policy base-policy rulebase-ips rule R1 then action drop-connection
set security idp idp-policy base-policy rulebase-ips rule R1 then notification log-attacks
  alert
set security idp idp-policy base-policy rulebase-ips rule R1 then severity critical
set security idp active-policy base-policy
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To define rules for an IDP IPS rulebase:

1. Create a policy by assigning a meaningful name to it.

```
[edit]
user@host# set security idp idp-policy base-policy
```

2. Associate a rulebase with the policy.

```
[edit security idp idp-policy base-policy]
user@host# set rulebase-ips
```

3. Add rules to the rulebase.

```
[edit security idp idp-policy base-policy rulebase-ips]
user@host# set rule R1
```

4. Define the match criteria for the rule.

```
[edit security idp idp-policy base-policy rulebase-ips rule R1]
user@host# set match from-zone trust to-zone untrust source-address any
  destination-address any application default
```

5. Define an attack as match criteria.

```
[edit security idp idp-policy base-policy rulebase-ips rule R1]
user@host# set match attacks predefined-attack-groups "Critical-TELNET"
```

6. Specify an action for the rule.

```
[edit security idp idp-policy base-policy rulebase-ips rule R1]
user@host# set then action drop-connection
```

7. Specify notification and logging options for the rule.

```
[edit security idp idp-policy base-policy rulebase-ips rule R1]
```

```
user@host# set then notification log-attacks alert
```

8. Set the severity level for the rule.

```
[edit security idp idp-policy base-policy rulebase-ips rule R1]
user@host# then severity critical
```

9. Activate the policy.

```
[edit]
user@host# set security idp active-policy base-policy
```

Results From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security idp
idp-policy base-policy {
  rulebase-ips {
    rule R1 {
      match {
        from-zone trust;
        source-address any;
        to-zone untrust;
        destination-address any;
        application default;
        attacks {
          predefined-attack-groups Critical-TELNET;
        }
      }
    }
  }
  then {
    action {
      drop-connection;
    }
    notification {
      log-attacks {
        alert;
      }
    }
    severity critical;
  }
}
active-policy base-policy;
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration on page 483

Verifying the Configuration

Purpose Verify if the rules for the IDP IPS rulebase configuration are correct

Action From operational mode, enter the **show security idp** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP IPS Rulebases on page 480
 - Example: Enabling IDP in a Security Policy on page 465
 - Example: Inserting a Rule in the IDP Rulebase on page 477
 - Example: Deactivating and Activating Rules in an IDP Rulebase on page 478

IDP Exempt Rulebase

- Understanding IDP Exempt Rulebases on page 484
- Example: Defining Rules for an IDP Exempt Rulebase on page 485

Understanding IDP Exempt Rulebases

The exempt rulebase works in conjunction with the intrusion prevention system (IPS) rulebase to prevent unnecessary alarms from being generated. You configure rules in this rulebase to exclude known false positives or to exclude a specific source, destination, or source/destination pair from matching an IPS rule. If traffic matches a rule in the IPS rulebase, the system attempts to match the traffic against the exempt rulebase before performing the action specified. Carefully written rules in an exempt rulebase can significantly reduce the number of false positives generated by an IPS rulebase.

Configure an exempt rulebase in the following conditions:

- When an IDP rule uses an attack object group that contains one or more attack objects that produce false positives or irrelevant log records.
- When you want to exclude a specific source, destination, or source/destination pair from matching an IDP rule. This prevents IDP from generating unnecessary alarms.



NOTE: Make sure to configure the IPS rulebase before configuring the exempt rulebase.

Table 47 on page 484 summarizes the options that you can configure in the exempt-rulebase rules.

Table 47: Exempt Rulebase Options

Term	Definition
Match condition	Specify the type of network traffic you want the device to monitor for attacks in the same way as in the IPS rulebase. However, in the exempt rulebase, you cannot configure an application; it is always set to any .
Attack objects/groups	Specify the attack objects that you do <i>not</i> want the device to match in the monitored network traffic.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP Policies Overview on page 463
 - Understanding IDP Policy Rules on page 470
 - Understanding IDP Policy Rulebases on page 476
 - Understanding IDP IPS Rulebases on page 480
 - Understanding Predefined IDP Policy Templates on page 537
 - Example: Defining Rules for an IDP Exempt Rulebase on page 485

Example: Defining Rules for an IDP Exempt Rulebase

This example shows how to define rules for an exempt IDP rulebase.

- Requirements on page 485
- Overview on page 485
- Configuration on page 485
- Verification on page 487

Requirements

Before you begin, create rules in the IPS rulebase. See “Example: Inserting a Rule in the IDP Rulebase” on page 477.

Overview

When you create an exempt rule, you must specify the following:

- Source and destination for traffic you want to exempt. You can set the source or destination to **Any** to exempt network traffic originating from any source or sent to any destination. You can also set **source-except** or **destination-except** to specify all the sources or destinations except the specified source or destination addresses.
- The attacks you want IDP to exempt for the specified source/destination addresses. You must include at least one attack object in an exempt rule.

This example shows that the IDP policy generates false positives for the attack **FTP:USER:ROOT** on an internal network. You configure the rule to exempt attack detection for this attack when the source IP is from your internal network.

Configuration

CLI Quick Configuration

To quickly define rules for an exempt IDP rulebase, copy the following commands and paste them into the CLI.

```
[edit]
set security idp idp-policy P1
set security idp idp-policy P1 rulebase-exempt rule R1 match from-zone trust to-zone any
set security idp idp-policy P1 rulebase-exempt rule R1 match source-address
  internal-devices destination-address any
set security idp idp-policy P1 rulebase-exempt rule R1 match attacks predefined-attacks
  "FTP:USER:ROOT"
set security idp active-policy P1
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To define rules for an exempt IDP rulebase:

1. Specify the IPS rulebase for which you want to define and exempt rulebase.

```
[edit]
user@host# set security idp idp-policy P1
```
2. Associate the exempt rulebase with the policy and zones, and add a rule to the rulebase.

```
[edit security idp idp-policy P1]
user@host# set rulebase-exempt rule R1 match from-zone trust to-zone any
```
3. Specify the source and destination addresses for the rulebase.

```
[edit security idp idp-policy P1]
user@host# set rulebase-exempt rule R1 match source-address internal-devices
destination-address any
```
4. Specify the attacks that you want to exempt from attack detection.

```
[edit security idp idp-policy P1]
user@host# set rulebase-exempt rule R1 match attacks predefined-attacks
"FTP:USER:ROOT"
```
5. Activate the policy.

```
[edit]
user@host# set security idp active-policy P1
```

Results From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security idp
idp-policy P1 {
  rulebase-exempt {
    rule R1 {
      match {
        from-zone trust;
        source-address internal-devices;
        to-zone any;
        destination-address any;
        attacks {
          predefined-attacks FTP:USER:ROOT;
        }
      }
    }
  }
}
active-policy P1;
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration on page 487

Verifying the Configuration

Purpose Verify if the defined rules were exempt from the IDP rulebase configuration.

Action From operational mode, enter the **show security idp** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Exempt Rulebases on page 484
 - Example: Inserting a Rule in the IDP Rulebase on page 477
 - Example: Deactivating and Activating Rules in an IDP Rulebase on page 478
 - Example: Enabling IDP in a Security Policy on page 465

IDP Terminal Rules

- Understanding IDP Terminal Rules on page 487
- Example: Setting Terminal Rules in Rulebases on page 488

Understanding IDP Terminal Rules

The Intrusion Detection and Prevention (IDP) rule-matching algorithm starts from the top of the rulebase and checks traffic against all rules in the rulebase that match the source, destination, and service. However, you can configure a rule to be *terminal*. A *terminal* rule is an exception to this algorithm. When a match is discovered in a terminal rule for the source, destination, zones, and application, IDP does not continue to check subsequent rules for the same source, destination, and application. It does not matter whether or not the traffic matches the attack objects in the matching rule.

You can use a terminal rule for the following purposes:

- To set different actions for different attacks for the same Source and Destination.
- To disregard traffic that originates from a known trusted source. Typically, the action is **None** for this type of terminal rule.
- To disregard traffic sent to a server that is vulnerable only to a specific set of attacks. Typically, the action is **Drop Connection** for this type of terminal rule.

Use caution when defining terminal rules. An inappropriate terminal rule can leave your network open to attacks. Remember that traffic matching the source, destination, and application of a terminal rule is not compared to subsequent rules, even if the traffic does not match an attack object in the terminal rule. Use a terminal rule only when you want to examine a certain type of traffic for one specific set of attack objects. Be particularly careful about terminal rules that use **any** for both the source and destination. Terminal rules should appear near the top of the rulebase before other rules that would match the same traffic.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP Policies Overview on page 463
 - Understanding IDP Policy Rules on page 470
 - Understanding IDP Policy Rulebases on page 476
 - Understanding IDP IPS Rulebases on page 480
 - Understanding IDP Exempt Rulebases on page 484
 - Example: Setting Terminal Rules in Rulebases on page 488

Example: Setting Terminal Rules in Rulebases

This example shows how to configure terminal rules.

- Requirements on page 488
- Overview on page 488
- Configuration on page 488
- Verification on page 489

Requirements

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Enable IDP application services in a security policy. See “Example: Enabling IDP in a Security Policy” on page 465.
- Create security zones. See “Example: Creating Security Zones” on page 88.
- Define rules. See “Example: Inserting a Rule in the IDP Rulebase” on page 477.

Overview

By default, rules in the IDP rulebase are not terminal. That means that IDP examines all rules in the rulebase and executes all matches. You can specify that a rule is terminal; if IDP encounters a match for the source, destination, and service specified in a terminal rule, it does not examine any subsequent rules for that connection.

This example shows how to configure terminal rules. You define a rule **R2** to terminate the match algorithm if the source IP of the traffic originates from a known trusted network in your company. If this rule is matched, IDP disregards traffic from the trusted network and does not monitor the session for malicious data.

Configuration

CLI Quick Configuration To quickly configure terminal rules, copy the following commands and paste them into the CLI.

```
[edit]
set security idp idp-policy P1 rulebase-ips rule R2
set security idp idp-policy P1 rulebase-ips rule R2 match source-address internal
destination-address any
```



```
set security idp idp-policy P1 rulebase-ips rule R2 terminal
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure terminal rules:

1. Define a rule and add it to a rulebase in a policy.

```
[edit]
user@host# set security idp idp-policy P1 rulebase-ips rule R2
```

2. Define the match criteria for the rule.

```
[edit security idp idp-policy P1]
user@host# set rulebase-ips rule R2 match source-address internal
destination-address any
```

3. Set the terminal flag for the rule.

```
[edit security idp idp-policy P1]
user@host# set rulebase-ips rule R2 terminal
```

Results From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security idp
idp-policy P1 {
  rulebase-ips {
    rule R2 {
      match {
        source-address internal;
        destination-address any;
      }
      terminal;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration on page 489

Verifying the Configuration

Purpose Verify if the terminal rules were configured.

Action From operational mode, enter the **show security idp** command.

Related Topics • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- [Understanding IDP Terminal Rules on page 487](#)
- [Example: Defining Rules for an IDP IPS Rulebase on page 481](#)
- [Example: Enabling IDP in a Security Policy on page 465](#)

IDP DSCP Rules

- [Understanding DSCP Rules in IDP Policies on page 490](#)
- [Example: Configuring DSCP Rules in an IDP Policy on page 490](#)

Understanding DSCP Rules in IDP Policies

Differentiated Services code point (DSCP) is an integer value encoded in the 6-bit field defined in IP packet headers. It is used to enforce class-of-service (CoS) distinctions. CoS allows you to override the default packet forwarding behavior and assign service levels to specific traffic flows.

You can configure DSCP value as an action in an IDP policy rule. You first define the traffic by defining match conditions in the IDP policy and then associate a DiffServ marking action with it. Based on the DSCP value, behavior aggregate classifiers set the forwarding class and loss priority for the traffic deciding the forwarding treatment the traffic receives.

All packets that match the IDP policy rule have the CoS field in their IP header rewritten with the DSCP value specified in the matching policy. If the traffic matches multiple rules with differing DSCP values, the first IDP rule that matches takes effect and this IDP rule then applies to all traffic for that session.

Related Topics

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [IDP Policies Overview on page 463](#)
- [Understanding IDP Policy Rules on page 470](#)
- [Understanding IDP Policy Rulebases on page 476](#)
- [Understanding IDP IPS Rulebases on page 480](#)
- [Understanding IDP Exempt Rulebases on page 484](#)
- [Example: Configuring DSCP Rules in an IDP Policy on page 490](#)

Example: Configuring DSCP Rules in an IDP Policy

This example shows how to configure DSCP values in an IDP policy.

- [Requirements on page 490](#)
- [Overview on page 491](#)
- [Configuration on page 491](#)
- [Verification on page 493](#)

Requirements

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Enable IDP application services in a security policy. See “Example: Enabling IDP in a Security Policy” on page 465.
- Create security zones. See “Example: Creating Security Zones” on page 88.
- Define rules. See “Example: Inserting a Rule in the IDP Rulebase” on page 477.

Overview

Configuring DSCP values in IDP policies provides a method of associating class-of-service (CoS) values—thus different levels of reliability—for different types of traffic on the network.

This example shows how to create a policy called **policy1**, specify a rulebase for this policy, and then add a rule **R1** to this rulebase. In this example, rule **R1**:

- Specifies the match condition to include any traffic from a previously configured zone called **zone1** to another previously configured zone called **zone2**. The match condition also includes a predefined attack group called **Critical - HTTP**. The application setting in the match condition is specified as **default** and matches any application configured in the attack object.
- Specifies an action to rewrite the CoS field in the IP header with the DSCP value **50** for any traffic that matches the criteria for rule **R1**,

Configuration

CLI Quick Configuration

To quickly configure DSCP values in an IDP policy, copy the following commands and paste them into the CLI.

```
[edit]
set security idp idp-policy policy1
set security idp idp-policy policy1 rulebase-ips rule R1
set security idp idp-policy P1 rulebase-ips rule R1 match from-zone Zone-1 to-zone Zone-2
  source-address any destination-address any application default
set security idp idp-policy P1 rulebase-ips rule R1 match attacks predefined-attack-groups
  "Critical-HTTP"
set security idp idp-policy P1 rulebase-ips rule R1 then action mark-diffserv 50
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure DSCP values in an IDP policy:

1. Create a policy by assigning a meaningful name to it.

```
[edit]
user@host# set security idp idp-policy policy1
```

2. Associate a rulebase with the policy.

```
[edit security idp idp-policy policy1]
```

```
user@host# set rulebase-ips
```

3. Add rules to the rulebase

```
[edit security idp idp-policy policy1 rulebase-ips]
user@host# set rule R1
```

4. Define the match criteria for the rule. The **default** application setting matches any application configured in the attack object.

```
[edit security idp idp-policy policy1 rulebase-ips R1]
user@host# set match from-zone zone1 to-zone zone2 source-address any
destination-address any application default
user@host# set match attacks predefined-attack-group "Critical - HTTP"
```

5. Specify an action for the rule.

```
[edit security idp idp-policy policy1 rulebase-ips R1]
user@host# set then action mark-diffserv 50
```

6. Continue to specify any notification or logging options for the rule, if required.

7. Activate the policy.

```
[edit]
user@host# set security idp active-policy policy1
```

Results From configuration mode, confirm your configuration by entering the **show security idp** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show security idp
idp-policy policy1 {
  rulebase-ips {
    rule R1 {
      match {
        from-zone Zone-1;
        source-address any;
        to-zone Zone-2;
        destination-address any;
        application default;
        attacks {
          predefined-attack-groups Critical-HTTP;
        }
      }
      then {
        action {
          mark-diffserv {
            50;
          }
        }
      }
    }
  }
}
active-policy policy1;
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration on page 493

Verifying the Configuration

Purpose Verify if the DSCP values were configured in an IDP policy.

Action From operational mode, enter the **show security idp** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding DSCP Rules in IDP Policies on page 490
 - Example: Enabling IDP in a Security Policy on page 465
 - Example: Defining Rules for an IDP IPS Rulebase on page 481

IDP Applications and Application Sets

- Understanding IDP Application Sets on page 493
- Example: Configuring IDP Applications and Services (CLI) on page 494
- Example: Configuring IDP Applications Sets (CLI) on page 495

Understanding IDP Application Sets

Applications or services represent Application Layer protocols that define how data is structured as it travels across the network. Because the services you support on your network are the same services that attackers must use to attack your network, you can specify which services are supported by the destination IP to make your rules more efficient. Juniper Networks provides predefined applications and application sets that are based on industry-standard applications. If you need to add applications that are not included in the predefined applications, you can create custom applications or modify predefined applications to suit your needs.

You specify an application, or service, to indicate that a policy applies to traffic of that type. Sometimes the same applications or a subset of them can be present in multiple policies, making them difficult to manage. Junos OS allows you to create groups of applications called *application sets*.

Application sets simplify the process by allowing you to manage a small number of application sets, rather than a large number of individual application entries.

The application (or application set) is configured as a match criterion for packets. Packets must be of the application type specified in the policy for the policy to apply to the packet. If the packet matches the application type specified by the policy and all other criteria match, then the policy action is applied to the packet. You can use predefined or custom applications and refer to them in a policy.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP Policies Overview on page 463
 - Understanding IDP Policy Rules on page 470
 - Understanding IDP Policy Rulebases on page 476
 - Example: Configuring IDP Applications and Services on page 494

Example: Configuring IDP Applications and Services (CLI)

To create custom applications, specify a meaningful name for an application and associate parameters with it—for example, inactivity timeout, or application protocol type.

Before you begin:

1. Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Enable Intrusion Detection and Prevention (IDP) application services in a security policy. See “Example: Enabling IDP in a Security Policy” on page 465.

The configuration instructions in this topic describe how to create an application **cust-app** and specify it as a match condition in the IDP policy **ABC**. In this example you create a special FTP application running on port **78**. You also specify the inactivity timeout value as 6000 seconds:

To create an application and associate it with an IDP policy:

1. Specify a unique name for the application. The following statement specifies **cust-app** as the name of the application:

```
user@host# set applications application cust-app
```
2. Specify application properties. The following statement specifies an FTP application using the TCP protocol and the port **78**. Inactivity timeout for the FTP service is set to 6000 seconds.

```
user@host# set applications application cust-app application-protocol ftp protocol tcp destination-port 78 inactivity-timeout 6000
```
3. Specify the application as a match condition in a policy. The following statement adds the **cust-app** application to the **ABC** policy:

```
user@host# set security idp idp-policy ABC rulebase-ips rule ABC match application cust-app
```
4. If you are finished configuring the device, commit the configuration.
5. For more information, see the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Application Sets on page 493

- Example: Configuring IDP Applications Sets on page 495
- Example: Enabling IDP in a Security Policy on page 465

Example: Configuring IDP Applications Sets (CLI)

To configure an application set, you add predefined or custom applications separately to an application set and assign a meaningful name to the application set. Once you name the application set you specify the name as part of the policy. For this policy to apply on a packet, the packet must match any one of the applications included in this set.

Before you begin:

1. Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Enable Intrusion Detection and Prevention (IDP) application services in a security policy. See “Example: Enabling IDP in a Security Policy” on page 465.

The configuration instructions in this topic describe how to create an application set **SrvAccessAppSet** and associate it with an IDP policy **ABC**. The application set **SrvAccessAppSet** combines three applications. Instead of specifying three applications in the policy rule, you specify one application set. If all of the other criteria match, any one of the applications in the application set serves as valid matching criteria.

To create an application set and associate it with an IDP policy:

1. Create an application set and specify applications to be included in the set. The following statements create the **SrvAccessAppSet** application set that includes a set of three applications:


```
user@host# set applications application-set SrvAccessAppSet application ssh
user@host# set applications application-set SrvAccessAppSet application telnet
user@host# set applications application-set SrvAccessAppSet application custApp
```
2. Associate the application set with an IDP policy. The following statement associates the application set **SrvAccessAppSet** to IDP policy **ABC**:


```
user@host# set security idp idp-policy ABC rulebase-ips rule ABC match application
SrvAccessAppSet
```
3. Specify an action for the policy. The following statement permits traffic from applications specified in the application set:


```
user@host# set security idp idp-policy ABC rulebase-ips rule ABC then action
no-action
```
4. If you are finished configuring the device, commit the configuration.
5. For more information, see the *Junos OS CLI Reference*.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding IDP Application Sets on page 493

- Example: Configuring IDP Applications and Services on page 494
- Example: Enabling IDP in a Security Policy on page 465

IDP Attacks and Attack Objects

- Understanding Custom Attack Objects on page 496
- IDP Protocol Decoders on page 512
- IDP Signature-Based Attacks on page 514
- IDP Protocol Anomaly-Based Attacks on page 517
- Example: Specifying IDP Test Conditions for a Specific Protocol (CLI) on page 519

Understanding Custom Attack Objects

You can create custom attack objects to detect new attacks or customize predefined attack objects to meet the unique needs of your network.

To configure a custom attack object, you specify a unique name for it and then specify additional information, such as a general description and keywords, which can make it easier for you to locate and maintain the attack object.

Certain properties in the attack object definitions are common to all types of attacks, such as attack name, description, severity level, service or application binding, time binding, recommended action, and protocol or port binding. Some fields are specific to an attack type and are available only for that specific attack definition.

This topic includes the following sections:

- Attack Name on page 496
- Severity on page 496
- Service and Application Bindings on page 497
- Protocol and Port Bindings on page 500
- Time Bindings on page 502
- Attack Properties (Signature Attacks) on page 503
- Attack Properties (Protocol Anomaly Attacks) on page 508
- Attack Properties (Compound or Chain Attacks) on page 509

Attack Name

Specify an alphanumeric name for the object. You might want to include the protocol the attack uses in the attack name.

Severity

Specifies the brutality of the attack on your network. Severity categories, in order of increasing brutality, are info, warning, minor, major, critical (see “Understanding IDP Rule Notifications” on page 475). Critical attacks are the most dangerous—typically these attacks attempt to crash your server or gain control of your network. Informational attacks

are the least dangerous, and typically are used by network administrators to discover holes in their own security systems.

Service and Application Bindings

The service or application binding field specifies the service that the attack uses to enter your network.



NOTE: Specify either the service or the protocol binding in a custom attack. In case you specify both, the service binding takes precedence.

- **Any**—Specify **any** if you are unsure of the correct service and want to match the signature in all services. Because some attacks use multiple services to attack your network, you might want to select the **Any** service binding to detect the attack regardless of which service the attack chooses for a connection.
- **Service**—Most attacks use a specific service to attack your network. You can select the specific service used to perpetrate the attack as the service binding. Table 48 on page 497 displays supported services and default ports associated with the services.

Table 48: Supported Services for Service Bindings

Service	Description	Default Port
AIM	AOL Instant Messenger. America Online Internet service provider (ISP) provides Internet, chat, and instant messaging applications.	TCP/5190
BGP	Border Gateway Protocol	TCP/179
Chargen	Character Generator Protocol is a UDP- or TCP-based debugging and measurement tool.	TCP/19, UDP/19
DHCP	Dynamic Host Configuration Protocol allocates network addresses and delivers configuration parameters from server to hosts.	UDP/67, UDP/68
Discard	Discard protocol is an Application Layer protocol that describes a process for discarding TCP or UDP data sent to port 9.	TCP/9, UDP/9
DNS	Domain Name System translates domain names into IP addresses.	TCP/53, UDP/53
Echo	Echo	TCP/7, UDP/7
Finger	Finger is a UNIX program that provides information about users.	TCP/79, UDP/79
FTP	File Transfer Protocol (FTP) allows the sending and receiving of files between machines.	TCP/21, UDP/21

Table 48: Supported Services for Service Bindings (*continued*)

Service	Description	Default Port
Gnutella	Gnutella is a public domain file sharing protocol that operates over a distributed network.	TCP/6346
Gopher	Gopher organizes and displays Internet servers' contents as a hierarchically structured list of files.	TCP/70
H225RAS	H.225.0/RAS (Registration, Admission, and Status)	UDP/1718, UDP/1719
HTTP	HyperText Transfer Protocol is the underlying protocol used by the World Wide Web (WWW).	TCP/80, TCP/81, TCP/88, TCP/3128, TCP/7001 (Weblogic), TCP/8000, TCP/8001, TCP/8100 (JRun), TCP/8200 (JRun), TCP/8080, TCP/8888 (Oracle-9i), TCP/9080 (Websphere), UDP/80
ICMP	Internet Control Message Protocol	
IDENT	Identification protocol is a TCP/IP Application Layer protocol used for TCP client authentication.	TCP/113
IKE	Internet Key Exchange protocol (IKE) is a protocol to obtain authenticated keying material for use with ISAKMP.	UDP/500
IMAP	Internet Message Access Protocol is used for retrieving messages.	TCP/143, UDP/143
IRC	Internet Relay Chat (IRC) allows people connected to the Internet to join live discussions.	TCP/6667
LDAP	Lightweight Directory Access Protocol is a set of protocols used to access information directories.	TCP/389
lpr	Line Printer Daemon protocol is a TCP-based protocol used for printing applications.	TCP/515
MSN	Microsoft Network Messenger is a utility that allows you to send instant messages and talk online.	TCP/1863
MSRPC	Microsoft Remote Procedure Call	TCP/135, UDP/135
MSSQL	Microsoft SQL is a proprietary database server tool that allows for the creation, access, modification, and protection of data.	TCP/1433, TCP/3306
MYSQL	MySQL is a database management system available for both Linux and Windows.	TCP/3306

Table 48: Supported Services for Service Bindings (*continued*)

Service	Description	Default Port
NBDS	NetBIOS Datagram Service application, published by IBM, provides connectionless (datagram) applications to PCs connected with a broadcast medium to locate resources, initiate sessions, and terminate sessions. It is unreliable and the packets are not sequenced.	UDP/137 (NBName), UDP/138 (NBDS)
NFS	Network File System uses UDP to allow network users to access shared files stored on computers of different types. SUN RPC is a building block of NFS.	TCP/2049, UDP/2049
nntp	Network News Transport Protocol is a protocol used to post, distribute, and retrieve USENET messages.	TCP/119
NTP	Network Time Protocol provides a way for computers to synchronize to a time reference.	UDP/123
POP3	Post Office Protocol is used for retrieving e-mail.	UDP/110, TCP/110
Portmapper	Service that runs on nodes on the Internet to map an ONC RPC program number to the network address of the server that listens for the program number.	TCP/111, UDP/111
RADIUS	Remote Authentication Dial-In User Service application is a server program used for authentication and accounting purposes.	UDP/1812, UDP/1813
rexec	Rexec	TCP/512
rlogin	RLOGIN starts a terminal session on a remote host.	TCP/513
rsh	RSH executes a shell command on a remote host.	TCP/514
rtsp	Real-Time Streaming Protocol (RTSP) is for streaming media applications	TCP/554
SIP	Session Initiation Protocol (SIP) is an Application-Layer control protocol for creating, modifying, and terminating sessions.	TCP/5060, UDP/5060
SMB	Server Message Block (SMB) over IP is a protocol that allows you to read and write files to a server on a network.	TCP/139, TCP/445
SMTP	Simple Mail Transfer Protocol is used to send messages between servers.	TCP/25, UDP/25
SNMP	Simple Network Management Protocol is a set of protocols for managing complex networks.	TCP/161, UDP/161
SNMPTRAP	SNMP trap	TCP/162, UDP/162

Table 48: Supported Services for Service Bindings (*continued*)

Service	Description	Default Port
SQLMON	SQL monitor (Microsoft)	UDP/1434
SSH	SSH is a program to log into another computer over a network through strong authentication and secure communications on a channel that is not secure.	TCP/22, UDP/22
SSL	Secure Sockets Layer	TCP/443, TCP/80
syslog	Syslog is a UNIX program that sends messages to the system logger.	UDP/514
Telnet	Telnet is a UNIX program that provides a standard method of interfacing terminal routers and terminal-oriented processes to each other.	TCP/23, UDP/23
TNS	Transparent Network Substrate	TCP/1521, TCP/1522, TCP/1523, TCP/1524, TCP/1525, TCP/1526, TCP/1527, TCP/1528, TCP/1529, TCP/1530, TCP/2481, TCP/1810, TCP/7778
TFTP	Trivial File Transfer Protocol	UDP/69
VNC	Virtual Network Computing facilitates viewing and interacting with another computer or mobile router connected to the Internet.	TCP/5800, TCP/5900
Whois	Network Directory Application Protocol is a way to look up domain names.	TCP/43
YMSG	Yahoo! Messenger is a utility that allows you to check when others are online, send instant messages, and talk online.	TCP/5050

Protocol and Port Bindings

Protocol or port bindings allow you to specify the protocol that an attack uses to enter your network. You can specify the name of the network protocol, or the protocol number.



NOTE: Specify either the service or the protocol binding in a custom attack. In case you specify both, the service binding takes precedence.

- **IP**—You can specify any of the supported network layer protocols using protocol numbers. Table 49 on page 501 lists protocol numbers for different protocols.

Table 49: Supported Protocols and Protocol Numbers

Protocol Name	Protocol Number
IGMP	2
IPIP	4
EGP	8
PUP	12
TP	29
IPV6	41
ROUTING	43
FRAGMENT	44
RSVP	46
GRE	47
ESP	50
AH	51
ICMPV6	58
NONE	59
DSTOPTS	60
MTP	92
ENCAP	98
PIM	103
COMP	108
RAW	255

- ICMP, TCP, and UDP—Attacks that do not use a specific service might use specific ports to attack your network. Some TCP and UDP attacks use standard ports to enter your network and establish a connection.
- RPC—The remote procedure call (RPC) protocol is used by distributed processing applications to handle interaction between processes remotely. When a client makes a remote procedure call to an RPC server, the server replies with a remote program;

each remote program uses a different program number. To detect attacks that use RPC, configure the service binding as RPC and specify the RPC program ID.

Table 50 on page 502 displays sample formats for key protocols.

Table 50: Sample Formats for Protocols

Protocol Name	Protocol Number	Description
ICMP	<code><Port>ICMP</Port></code>	Specify the protocol name.
IP	<code><Port>IP/protocol-number</Port></code>	Specify the Network Layer protocol number.
RPC	<code><Port>RPC/program-number</Port></code>	Specify the RPC program number.
TCP or UDP	<ul style="list-style-type: none"> <code><Port>TCP </Port></code> <code><Port>TCP/port </Port></code> <code><Port>TCP/minport-maxport </Port></code> 	Specifying the port is optional for TCP and UDP protocols. For example, you can specify either of the following: <ul style="list-style-type: none"> <code><Port>UDP</Port></code> <code><Port>UDP/10</Port></code> <code><Port>UDP/10-100</Port></code>

Time Bindings

Use time bindings to configure the time attributes for the custom attack object. Time attributes control how the attack object identifies attacks that repeat for a certain number of times. By configuring the scope and count of an attack, you can detect a sequence of the same attacks over a period of time (one minute) across sessions.

Scope

Specify the scope within which the count of an attack occurs:

- **Source**—Specify this option to detect attacks from the source address for the specified number of times, regardless of the destination address. This means that for a given attack, a threshold value is maintained for each attack from the source address. The destination address is ignored. For example, anomalies are detected from two different pairs (**ip-a**, **ip-b**) and (**ip-a**, **ip-c**) that have the same source address **ip-a** but different destination addresses **ip-b** and **ip-c**. Then the number of matches for **ip-a** increments to 2. Suppose the threshold value or *count* is also set to 2, then the signature triggers the attack event.
- **Destination**—Specify this option to detect attacks sent to the destination address for the specified number of times, regardless of the source address. This means that for a given attack, a threshold value is maintained for each attack from the destination address. The source address is ignored. For example, if anomalies are detected from two different pairs (**ip-a**, **ip-b**) and (**ip-c**, **ip-b**) that have the same destination address **ip-b** but different source addresses **ip-a** and **ip-c**. Then the number of matches for **ip-b** increments to 2. Suppose the threshold value or *count* is also set to 2, then the signature triggers the attack event.

- **Peer**—Specify this option to detect attacks between source and destination IP addresses of the sessions for the specified number of times. This means that the threshold value is applicable for a pair of source and destination addresses. Suppose anomalies are detected from two different source and destination pairs (**ip-a, ip-b**) and (**ip-a, ip-c**). Then the number of matches for each pair is set to 1, even though both pairs have a common source address.

Count

Count or threshold value specifies the number of times that the attack object must detect an attack within the specified scope before the device considers the attack object to match the attack. If you bind the attack object to multiple ports and the attack object detects that attack on different ports, each attack on each port is counted as a separate occurrence. For example, when the attack object detects an attack on **TCP/80** and then on **TCP/8080**, the count is two.

Once the **count** match is reached, each attack that matches the criteria causes the attack count to increase by one. This count cycle lasts for a duration of 60 seconds, after which the cycle repeats.

Attack Properties (Signature Attacks)

Signature attack objects use a stateful attack signature (a pattern that always exists within a specific section of the attack) to detect known attacks. They also include the protocol or service used to perpetrate the attack and the context in which the attack occurs. The following properties are specific to signature attacks, and you can configure them when configuring signature attack:



NOTE: Attack context, flow type, and direction are mandatory fields for the signature attack definition.

Attack Context

An attack context defines the location of the signature. If you know the service and the specific service context, specify that service and then specify the appropriate service contexts. If you know the service, but are unsure of the specific service context, specify one of the following general contexts:

- **first-data-packet**—Specify this context to detect the attack in only the first data packet.
- **first-packet**—Specify this context to detect the attack in only the first packet of a stream. When the flow direction for the attack object is set to **any**, the device checks the first packet of both the server-to-client and the client-to-server flows. If you know that the attack signature appears in the first packet of a session, choosing **first packet** instead of **packet** reduces the amount of traffic the device needs to monitor, which improves performance.
- **packet**—Specify this context to match the attack pattern within a packet. When you select this option, you must also specify the service binding to define the service header options. Although not required, specifying these additional parameters improves the accuracy of the attack object and thereby improves performance.

- **line**—Specify this context to detect a pattern match within a specific line within your network traffic.
- **normalized-stream**—Specify this context to detect the attack in an entire normalized stream. The normalized stream is one of the multiple ways of sending information. In this stream the information in the packet is normalized before a match is performed. Suppose **www.yahoo.com/sports** is the same as **www.yahoo.com/s%70orts**. The normalized form to represent both of these URLs might be **www.yahoo.com/sports**. Choose **normalized stream** instead of **stream**, unless you want to detect some pattern in its exact form. For example, if you want to detect the exact pattern **www.yahoo.com/s%70orts**, then select **stream**.
- **normalized-stream256**—Specify this context to detect the attack in only the first 256 bytes of a normalized stream.
- **normalized-stream1k**—Specify this context to detect the attack in only the first 1024 bytes of a normalized stream.
- **normalized-stream-8k**—Specify this context to detect the attack in only the first 8192 bytes of a normalized stream.
- **stream**—Specify this context to reassemble packets and extract the data to search for a pattern match. However, the device cannot recognize packet boundaries for stream contexts, so data for multiple packets is combined. Specify this option only when no other context option contains the attack.
- **stream256**—Specify this context to reassemble packets and search for a pattern match within the first 256 bytes of a traffic stream. When the flow direction is set to **any**, the device checks the first 256 bytes of both the server-to-client and client-to-server flows. If you know that the attack signature will appear in the first 256 bytes of a session, choosing **stream256** instead of **stream** reduces the amount of traffic that the device must monitor and cache, thereby improving performance.
- **stream1k**—Specify this context to reassemble packets and search for a pattern match within the first 1024 bytes of a traffic stream. When the flow direction is set to **any**, the device checks the first 1024 bytes of both the server-to-client and client-to-server flows. If you know that the attack signature will appear in the first 1024 bytes of a session, choosing **stream1024** instead of **stream** reduces the amount of traffic that the device must monitor and cache, thereby improving performance.
- **stream8k**—Specify this context to reassemble packets and search for a pattern match within the first 8192 bytes of a traffic stream. When the flow direction is set to **any**, the device checks the first 8192 bytes of both the server-to-client and client-to-server flows. If you know that the attack signature will appear in the first 8192 bytes of a session, choosing **stream8192** instead of **stream** reduces the amount of traffic that the device must monitor and cache, thereby improving performance.

Attack Direction

You can specify the connection direction of the attack. Using a single direction (instead of **Any**) improves performance, reduces false positives, and increases detection accuracy.

- Client to server (detects the attack only in client-to-server traffic)
- Server to client (detects the attack only in server-to-client traffic)

- Any (detects the attack in either direction)

Attack Pattern

Attack patterns are signatures of the attacks you want to detect. A signature is a pattern that always exists within an attack; if the attack is present, so is the signature. To create the attack pattern, you must first analyze the attack to detect a pattern (such as a segment of code, a URL, or a value in a packet header), then create a syntactical expression that represents that pattern. You can also negate a pattern. Negating a pattern means that the attack is considered matched if the pattern defined in the attack does *not* match the specified pattern.



NOTE: Pattern negation is supported for packet, line, and application based contexts only and not for stream and normalized stream contexts.

Protocol-Specific Parameters

Specifies certain values and options existing within packet headers. These parameters are different for different protocols. In a custom attack definition, you can specify fields for only one of the following protocols—TCP, UDP, or ICMP. Although, you can define IP protocol fields with TCP or UDP in a custom attack definition.



NOTE: Header parameters can be defined only for attack objects that use a packet or first packet context. If you specified a line, stream, stream 256, or a service context you cannot specify header parameters.

If you are unsure of the options or flag settings for the malicious packet, leave all fields blank and Intrusion Detection and Prevention (IDP) attempts to match the signature for all header contents.

Table 51 on page 505 displays fields and flags that you can set for attacks that use the IP protocol.

Table 51: IP Protocol Fields and Flags

Field	Description
Type of Service	Specify a value for the service type. Common service types are: <ul style="list-style-type: none"> • 0000 Default • 0001 Minimize Cost • 0002 Maximize Reliability • 0003 Maximize Throughput • 0004 Minimize Delay • 0005 Maximize Security
Total Length	Specify a value for the number of bytes in the packet, including all header fields and the data payload.

Table 51: IP Protocol Fields and Flags (*continued*)

Field	Description
ID	Specify a value for the unique value used by the destination system to reassemble a fragmented packet.
Time to Live	Specify an integer value in the range of 0–255 for the time-to-live (TTL) value of the packet. This value represents the number of devices the packet can traverse. Each router that processes the packet decrements the TTL by 1; when the TTL reaches 0, the packet is discarded.
Protocol	Specify a value for the protocol used.
Source	Enter the source address of the attacking device.
Destination	Enter the destination address of the attack target.
Reserved Bit	This bit is not used.
More Fragments	When set (1), this option indicates that the packet contains more fragments. When unset (0), it indicates that no more fragments remain.
Don't Fragment	When set (1), this option indicates that the packet cannot be fragmented for transmission.

Table 52 on page 506 displays packet header fields and flags that you can set for attacks that use the TCP protocol.

Table 52: TCP Header Fields and Flags

Field	Description
Source Port	Specify a value for the port number on the attacking device.
Destination Port	Specify a value for the port number of the attack target.
Sequence Number	Specify a value for the sequence number of the packet. This number identifies the location of the data in relation to the entire data sequence.
ACK Number	Specify a value for the ACK number of the packet. This number identifies the next sequence number; the ACK flag must be set to activate this field.
Header Length	Specify a value for the number of bytes in the TCP header.
Data Length	Specify a value for the number of bytes in the data payload. For SYN, ACK, and FIN packets, this field should be empty.
Window Size	Specify a value for the number of bytes in the TCP window size.

Table 52: TCP Header Fields and Flags (*continued*)

Field	Description
Urgent Pointer	Specify a value for the urgent pointer. The value indicates that the data in the packet is urgent; the URG flag must be set to activate this field.
URG	When set, the urgent flag indicates that the packet data is urgent.
ACK	When set, the acknowledgment flag acknowledges receipt of a packet.
PSH	When set, the push flag indicates that the receiver should push all data in the current sequence to the destination application (identified by the port number) without waiting for the remaining packets in the sequence.
RST	When set, the reset flag resets the TCP connection, discarding all packets in an existing sequence.
SYN	When set, the SYN flag indicates a request for a new session.
FIN	When set, the final flag indicates that the packet transfer is complete and the connection can be closed.
R1	This reserved bit (1 of 2) is not used.
R2	This reserved bit (2 of 2) is not used.

Table 53 on page 507 displays packet header fields and flags that you can set for attacks that use the UDP protocol.

Table 53: UDP Header Fields and Flags

Field	Description
Source Port	Specify a value for the port number on the attacking device.
Destination Port	Specify a value for the port number of the attack target.
Data Length	Specify a value for the number of bytes in the data payload.

Table 54 on page 507 displays packet header fields and flags that you can set for attacks that use the ICMP protocol.

Table 54: ICMP Header Fields and Flags

Field	Description
ICMP Type	Specify a value for the primary code that identifies the function of the request or reply packet.

Table 54: ICMP Header Fields and Flags (*continued*)

Field	Description
ICMP Code	Specify a value for the secondary code that identifies the function of the request or reply packet within a given type.
Sequence Number	Specify a value for the sequence number of the packet. This number identifies the location of the request or reply packet in relation to the entire sequence.
ICMP ID	Specify a value for the identification number. The identification number is a unique value used by the destination system to associate request and reply packets.
Data Length	Specify a value for the number of bytes in the data payload.

Sample Signature Attack Definition

The following is a sample signature attack definition:

```

<Entry>
<Name>sample-sig</Name>
<Severity>Major</Severity>
<Attacks><Attack>
<TimeBinding><Count>2</Count>
<Scope>dst</Scope></TimeBinding>
<Application>FTP</Application>
<Type>signature</Type>
<Context>packet</Context>
<Negate>true</Negate>
<Flow>Control</Flow>
<Direction>any</Direction>
<Headers><Protocol><Name>ip</Name>
<Field><Name>ttl</Name>
<Match>==</Match><Value>128</Value></Field>
</Protocol><Name>tcp</Name>
<Field><Name><Match>&lt;</Match>
<value>1500</Value>
</Field></Protocol></Headers>
</Attack></Attacks>
</Entry>

```

Attack Properties (Protocol Anomaly Attacks)

A protocol anomaly attack object detects unknown or sophisticated attacks that violate protocol specifications (RFCs and common RFC extensions). You cannot create new protocol anomalies, but you can configure a new attack object that controls how your device handles a predefined protocol anomaly when detected.



NOTE: The service or application binding is a mandatory field for protocol anomaly attacks.

The following properties are specific to protocol anomaly attacks. Both attack direction and test condition are mandatory fields for configuring anomaly attack definitions.

Attack Direction

Attack direction allows you to specify the connection direction of an attack. Using a single direction (instead of **Any**) improves performance, reduces false positives, and increases detection accuracy:

- Client to server (detects the attack only in client-to-server traffic)
- Server to client (detects the attack only in server-to-client traffic)
- Any (detects the attack in either direction)

Test Condition

Test condition is a condition to be matched for an anomaly attack. Juniper Networks supports certain predefined test conditions. In the following example, the condition is a message that is too long. If the size of the message is longer than the preconfigured value for this test condition, the attack is matched.

```
<Attacks>
<Attack>
<Type>anomaly</Type>
...
<Test>MESSAGE_TOO_LONG</Test>
<Value>yes</Value>
...
</Attack>
</Attacks>
```

Sample Protocol Anomaly Attack Definition

The following is a sample protocol anomaly attack definition:

```
<Entry>
<Name>sample-anomaly</Name>
<Severity>Info</Severity>
<Attacks><Attack>
<TimeBinding><Count>2</Count>
<Scope>peer</Scope></TimeBinding>
<Application>TCP</Application>
<Type>anomaly</Type>
<Test>OPTIONS_UNSUPPORTED</Test>
<Direction>any</Direction>
</Attack></Attacks>
</Entry>
```

Attack Properties (Compound or Chain Attacks)

A compound or chain attack object detects attacks that use multiple methods to exploit a vulnerability. This object combines multiple signatures and/or protocol anomalies into a single attack object, forcing traffic to match a pattern of combined signatures and anomalies within the compound attack object before traffic is identified as an attack. By combining and even specifying the order in which signatures or anomalies must match,

you can be very specific about the events that need to take place before the device identifies traffic as an attack.

You must specify a minimum of 2 members (attacks) in a compound attack. You can specify up to 32 members in compound attack. Members can be either signature or anomaly attacks.

The following properties are specific to compound attacks:

Scope

Scope allows you to specify if the attack is matched within a session or across transactions in a session. If the specified service supports multiple transactions within a single session, you can also specify whether the match should occur over a single session or can be made across multiple transactions within a session:

- Specify *session* to allow multiple matches for the object within the same session.
- Specify *transaction* to match the object across multiple transactions that occur within the same session.

Order

Use ordered match to create a compound attack object that must match each member signature or protocol anomaly in the order you specify. If you do not specify an ordered match, the compound attack object still must match all members, but the attack pattern or protocol anomalies can appear in the attack in random order.

Reset

Specifies that a new log is generated each time an attack is detected within the same session. If this field is set to **no** then the attack is logged only once for a session.

Expression (Boolean expression)

Using the boolean expression field disables the ordered match function. The boolean expression field makes use of the member name or member index properties. The following three boolean operators are supported along with parenthesis, which helps determine precedence:

- **or**—If either of the member name patterns match, the expression matches.
- **and**—If both of the member name patterns match, the expression matches. It does not matter which order the members appear in.
- **oand (ordered and)**—If both of the member name patterns match, and if they appear in the same order as specified in the boolean expression, the expression matches.

Suppose you have created six signature members, labelled **s1-s5**. Suppose you know that the attack always contains the pattern **s1**, followed by either **s2** or **s3**. You also know that the attack always contains **s4** and **s5**, but their positions in the attack can vary. In this case, you might create the following boolean expression: **((s1 oand s2) or (s1 oand s3)) and (s4 and s5)**



NOTE: You can either define an ordered match or an expression (not both) in a custom attack definition.

Member Index

Member Index is specified in chain attacks to identify a member (attack) uniquely. In the following example, member index is used to identify the members **m01** and **m02** in the defined expression:

```
<Expression>m02 AND m01</Expression>
<Order>no</Order>
<Reset>no</Reset>
<ScopeOption/>
<Members>
<Attack>
<Member>m01</Member>
<Type>Signature</Type>
...
<Pattern><![CDATA[.*/getlatestversion]]></Pattern>
<Regex/>
</Attack>
<Attack><Member>m02</Member>
<Type>Signature</Type>
...
<Pattern><![CDATA[\\[Skype\\'.*]]></Pattern>
<Regex/>
</Attack>
<Attack>
```



NOTE: When defining the expression, you must specify the member index for all members.

Sample Compound Attack Definition

The following is a sample compound attack definition:

```
<Entry>
<Name>sample-chain</Name>
<Severity>Critical</Severity>
<Attacks><Attack>
<Application>HTTP</Application>
<Type>Chain</Type>
<Order>yes</Order>
<Reset>yes</Reset>
<Members><Attack>
<Type>Signature</Type>
<Context>packet</Context>
<Pattern><![CDATA[Unknown[]]]></Pattern>
<Flow>Control</Flow>
<Direction>cts</Direction>
</Attack><Attack>
<Type>anomaly</Type>
<Test>CHUNK_LENGTH_OVERFLOW</Test>
```

```
<Direction>any</Direction>
</Attack></Members>
</Attack></Attacks>
</Entry>
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP Policies Overview on page 463
 - Understanding IDP Policy Rulebases on page 476
 - Understanding Predefined IDP Attack Objects and Object Groups on page 540
 - Understanding IDP Protocol Decoders on page 512
 - Understanding IDP Signature-Based Attacks on page 514
 - Understanding IDP Protocol Anomaly-Based Attacks on page 517

IDP Protocol Decoders

- Understanding IDP Protocol Decoders on page 512
- Example: Configuring IDP Protocol Decoders (CLI) on page 513
- Understanding Multiple IDP Detector Support on page 513

Understanding IDP Protocol Decoders

Protocol decoders are used by Intrusion Detection and Prevention (IDP) to check protocol integrity and protocol contextual information by looking for anomalies and ensuring that RFC standards are met. An anomaly can be any part of a protocol, such as the header, message body, or other individual fields that deviate from RFC standards for that protocol. For example, in the case of SMTP, if SMTP MAIL TO precedes SMTP HELO, that is an anomaly in the SMTP protocol.

When protocol contextual information is available, protocol decoders check for attacks within those contexts. For example, for SMTP, if an e-mail is sent to user@company.com, user@company.com is the contextual information and SMTP MAIL TO is the context. By using protocol contextual data, rather than the entire packet, for attack detection, protocol decoders improve overall performance and accuracy.

If there is a policy configured with a rule that matches the protocol decoder check for SMTP, the rule triggers and the appropriate action is taken.

The IDP module ships with a preconfigured set of protocol decoders. These protocol decoders have default settings for various protocol-specific contextual checks they perform. You can use these defaults or you can tune them to meet your site's specific needs. To display the list of available protocol decoders, enter the following command:

```
user@host # show security idp sensor-configuration detector protocol-name ?
```

For a more detailed view of the current set of protocol decoders and their default context values, you can view the *detector-capabilities.xml* file located in the `/var/db/idpd/sec-download` folder on the device. When you download a new security package, you also receive this file which lists current protocols and default decoder context values.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP Policies Overview on page 463
 - Understanding Custom Attack Objects on page 496
 - Understanding IDP Protocol Anomaly-Based Attacks on page 517
 - Understanding Multiple IDP Detector Support on page 513
 - Understanding IDP Signature-Based Attacks on page 514
 - Example: Configuring IDP Protocol Decoders on page 513

Example: Configuring IDP Protocol Decoders (CLI)

The configuration instructions in this topic provide a tunable context configuration example for one protocol decoder, FTP.

To configure protocol decoder tunables, refer to the following information:

1. View the list of protocols that have tunable parameters by entering the following command.

```
user@host # set security idp sensor-configuration detector protocol-name
```

2. To configure tunable parameters for the protocol in question (in this case, FTP), enter the following:

```
user@host # set security idp sensor-configuration detector protocol-name ftp
tunable-name ftp_failed_logins tunable-value 4
user@host # set security idp sensor-configuration detector protocol-name ftp
tunable-name ftp_failed_flags tunable value 1
user@host # set security idp sensor-configuration detector protocol-name ftp
tunable-name ftp_line_length tunable-value 1024
user@host # set security idp sensor-configuration detector protocol-name ftp
tunable-name ftp_password_length tunable-value 64
user@host # set security idp sensor-configuration detector protocol-name ftp
tunable-name ftp_sitestring_length tunable-value 512
user@host # set security idp sensor-configuration detector protocol-name ftp
tunable-name ftp_username_length tunable-value 32
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Protocol Decoders on page 512
 - Understanding Multiple IDP Detector Support on page 513
 - Understanding IDP Signature-Based Attacks on page 514

Understanding Multiple IDP Detector Support

When a new security package is received, it contains attack definitions and a detector. In any given version of a security package, the attack definitions correspond to the capabilities of the included detector. When policy aging is disabled on the device (see the *reset-on-policy* command in the *Junos OS CLI Reference* for policy aging commands), only one policy is in effect at any given time. But if policy aging is enabled and there is a policy update, the existing policy is not unloaded when the new policy is loaded. Therefore,

both policies can be in effect on the device. In this case, all existing sessions will continue to be inspected by existing policies and new sessions are inspected with new policies. Once all the existing sessions using the older policy have terminated or expired, the older policy is then unloaded.

When a policy is loaded, it is also associated with a detector. If the new policy being loaded has an associated detector that matches the detector already in use by the existing policy, the new detector is not loaded and both policies use a single associated detector. But if the new detector does not match the current detector, the new detector is loaded along with the new policy. In this case, each loaded policy will then use its own associated detector for attack detection.

Note that a maximum of two detectors can be loaded at any given time. If two detectors are already loaded (by two or more policies), and loading a new policy requires also loading a new detector, then at least one of the loaded detectors must be unloaded before the new detector can be loaded. Before a detector is unloaded, all policies that use the corresponding detector are unloaded as well.

You can view the current policy and corresponding detector version by entering the following command:

```
user@host> show security idp status
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Protocol Decoders on page 512
 - Example: Configuring IDP Protocol Decoders on page 513
 - Understanding IDP Signature-Based Attacks on page 514

IDP Signature-Based Attacks

- Understanding IDP Signature-Based Attacks on page 514
- Example: Configuring IDP Signature-Based Attacks (CLI) on page 515

Understanding IDP Signature-Based Attacks

To configure a custom attack object, you specify a unique name for it and then specify additional information, which can make it easier for you to locate and maintain the attack object.

Certain properties in the attack object definitions are common to all types of attacks, such as attack name, severity level, service or application binding, time binding, and protocol or port binding. Some fields are specific to an attack type and are available only for that specific attack definition.

Signature attack objects use a stateful attack signature (a pattern that always exists within a specific section of the attack) to detect known attacks. They also include the protocol or service used to perpetrate the attack and the context in which the attack occurs. The following properties are specific to signature attacks, and you can configure them when configuring signature attack—attack context, attack direction, attack pattern, and protocol-specific parameters (TCP, UDP, ICMP, or IP header fields).

When configuring signature-based attacks, keep the following in mind:

- Attack context and direction are mandatory fields for the signature attack definition.
- Pattern negation is supported for packet, line, and application-based contexts only and not for stream and normalized stream contexts.
- When configuring the protocol-specific parameters, you can specify fields for only one of the following protocols—IP, TCP, UDP, or ICMP.
- When configuring a protocol binding, you can specify only one of the following—IP, ICMP, TCP, UDP, RPC or applications.
 - IP—Protocol number is a mandatory field.
 - TCP and UDP—You can specify either a single port (minimum-port) or a port range (minimum-port and maximum-port). If you do not specify a port, the default value is taken (**0-655325**).
 - RPC—Program number is a mandatory field.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP Policies Overview on page 463
- Understanding Custom Attack Objects on page 496
- Understanding Predefined IDP Attack Objects and Object Groups on page 540
- Understanding IDP Protocol Decoders on page 512
- Example: Configuring IDP Signature-Based Attacks on page 515
- Example: Configuring IDP Protocol Anomaly-Based Attacks on page 517

Example: Configuring IDP Signature-Based Attacks (CLI)

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

The configuration instructions in this topic describe how to create a signature-based attack object. In this example, you create a signature attack named **sig1** and assign it the following properties:

- Recommended action (**drop packet**)—Specify to drop a matching packet before it can reach its destination but does not close the connection.
- Time binding—Specify the scope as **source** and count as **10**. When scope is **source**, all attacks from the same source are counted, and when the number of attack reaches the count (**10**) specified, the attack is logged. In this example, every tenth attack from the same source is logged.
- Attack context (**packet**)—Specify to match the attack pattern within a packet.
- Attack direction (**any**)—Specify to detect the attack in both directions—client-to-server and server-to-client traffic.
- Protocol (**TCP**)—Specify time to live (TTL) value of **128**.

- Shellcode (**Intel**)—Set the flag to detect shellcode for Intel platforms.
- Protocol binding—Specify TCP protocol and ports **50** through **100**.

Once you have configured a signature-based attack object, you specify the attack as match criteria in an IDP policy rule. For more information, see “Example: Defining Rules for an IDP IPS Rulebase” on page 481.

To create a signature-based attack object:

1. Specify a name for the attack. The following statement specifies **sig1** as the name of the attack.

```
user@host# set security idp custom-attack sig1
```
2. Specify common properties for the attack. The following statements specify a recommended action to drop packets and define time binding with scope as **source** scope and count as **10**.

```
user@host# set security idp custom-attack sig1 recommended-action drop-packet
user@host# set security idp custom-attack sig1 time-binding scope source count 10
```
3. Specify the attack type and context. The following statement specifies the attack type **signature** and context **packet**.

```
user@host# set security idp custom-attack sig1 attack-type signature context packet
```
4. Specify the attack direction and the shellcode flag. The following statement specifies the attack direction **any** and sets the shellcode flag to **intel**.

```
user@host# set security idp custom-attack sig1 attack-type signature shellcode intel
```
5. Set the protocol and its fields. The following statement specifies the IP protocol and the TTL value **128**.

```
user@host# set security idp custom-attack sig1 attack-type signature protocol ip ttl
value 128 match equal
```
6. Specify the protocol binding and ports. The following statement specifies the TCP protocol and the port range from **50** through **100**.

```
user@host# set security idp custom-attack sig1 attack-type signature protocol-binding
tcp minimum-port 50 maximum-port 100
```
7. If you are finished configuring the device, commit the configuration.
8. From configuration mode in the CLI, enter the **show security idp** command to verify the configuration. For more information, see the *Junos OS CLI Reference*.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding IDP Signature-Based Attacks on page 514
- Understanding Custom Attack Objects on page 496
- Understanding Predefined IDP Attack Objects and Object Groups on page 540
- Understanding IDP Protocol Decoders on page 512
- Example: Configuring IDP Protocol Anomaly-Based Attacks on page 517

IDP Protocol Anomaly-Based Attacks

- Understanding IDP Protocol Anomaly-Based Attacks on page 517
- Example: Configuring IDP Protocol Anomaly-Based Attacks (CLI) on page 517

Understanding IDP Protocol Anomaly-Based Attacks

A protocol anomaly attack object detects unknown or sophisticated attacks that violate protocol specifications (RFCs and common RFC extensions). You cannot create new protocol anomalies, but you can configure a new attack object that controls how your device handles a predefined protocol anomaly when detected.

The following properties are specific to protocol anomaly attacks:

- Attack direction
- Test condition

When configuring protocol anomaly-based attacks, keep the following in mind:

- The service or application binding is a mandatory field for protocol anomaly attacks. Besides the supported applications, services also include IP, TCP, UDP, ICMP, and RPC.
- The attack direction and test condition properties are mandatory fields for configuring anomaly attack definitions.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP Policies Overview on page 463
- Understanding IDP Protocol Decoders on page 512
- Understanding Custom Attack Objects on page 496
- Understanding Predefined IDP Attack Objects and Object Groups on page 540
- Example: Configuring IDP Protocol Anomaly-Based Attacks on page 517

Example: Configuring IDP Protocol Anomaly-Based Attacks (CLI)

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

The configuration instructions in this topic describe how to create a signature-based attack object. In this example, you create a protocol anomaly attack named **anomaly1** and assign it the following properties:

- Time binding—Specify the scope as **peer** and count as **2** to detect anomalies between source and destination IP addresses of the sessions for the specified number of times.
- Severity (**info**)—Specify to provide information about any attack that matches the conditions.
- Attack direction (**any**)—Specify to detect the attack in both directions—client-to-server and server-to-client traffic.
- Service (**TCP**)—Specify to match attacks using the TCP service.

- Test condition (**OPTIONS_UNSUPPORTED**)—Specify to match certain predefined test conditions. In this example, the condition is to match if the attack includes unsupported options.
- Shellcode (**sparc**)—Set the flag to detect shellcode for Sparc platforms.

Once you have configured the protocol anomaly-based attack object, you specify the attack as match criteria in an Intrusion Detection and Prevention (IDP) policy rule. For more information, see “Example: Defining Rules for an IDP IPS Rulebase” on page 481.

To create a protocol anomaly-based attack object:

1. Specify a name for the attack. The following statement specifies **anomaly1** as the name of the attack.

```
user@host# set security idp custom-attack anomaly1
```
2. Specify common properties for the attack. The following statements specify an **info** severity level and a time binding with a scope type **peer** and count **2**.

```
user@host# set security idp custom-attack anomaly1 severity info
user@host# set security idp custom-attack anomaly1 time-binding scope peer count
2
```
3. Specify the attack type and test condition. The following statement specifies the attack type **anomaly** and test condition **UNSUPPORTED_OPTIONS**.

```
user@host# set security idp custom-attack anomaly1 attack-type anomaly test
UNSUPPORTED_OPTIONS
```
4. Specify other properties for the anomaly attack. The following statement specifies the service TCP and attack direction **any**, and sets the shellcode flag to **sparc** and specifies .

```
user@host# set security idp custom-attack sa attack-type anomaly service TCP
user@host# set security idp custom-attack sa attack-type anomaly direction any
user@host# set security idp custom-attack sa attack-type anomaly shellcode sparc
```
5. If you are finished configuring the device, commit the configuration.
6. From configuration mode in the CLI, enter the **show security idp** command to verify the configuration. For more information, see the *Junos OS CLI Reference*.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding IDP Protocol Anomaly-Based Attacks on page 517
- Example: Defining Rules for an IDP IPS Rulebase on page 481
- Example: Updating the IDP Signature Database Manually (CLI) on page 543
- Example: Updating the Signature Database Automatically (CLI) on page 545

Example: Specifying IDP Test Conditions for a Specific Protocol (CLI)

When configuring Intrusion Detection and Prevention (IDP) custom attacks, you can specify test conditions for a specific protocol. For example, to configure test conditions for ICMP:

1. List supported test conditions for ICMP and choose the one you want to configure. The supported test conditions are available in the CLI at the **[edit security idp custom-attack test1 attack-type anomaly]** hierarchy level.

```
user@host# set test icmp?
```

Possible completions:

```
<test>                                Protocol anomaly condition to be checked
```

```
ADDRESSMASK_REQUEST
DIFF_CHECKSUM_IN_RESEND
DIFF_CHECKSUM_IN_RESPONSE
DIFF_LENGTH_IN_RESEND
```

2. Configure the service for which you want to configure the test condition.

```
user@host# set service ICMP
```

3. Configure the test condition (specifying the protocol name is not required):

```
user@host# set test ADDRESSMASK_REQUEST
```

4. If you are finished configuring the device, commit the configuration.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Protocol Anomaly-Based Attacks on page 517
 - Example: Configuring IDP Protocol Anomaly-Based Attacks on page 517

Limitations of IDP

On an SRX Series or a J Series device, when defining IDP, be aware of the following limitations:

- On SRX3400, SRX3600, SRX5600, and SRX5800 devices, application-level distributed denial-of-service (application-level DDoS) detection does not work if two rules with different application-level DDoS applications process traffic going to a single destination application server. When setting up application-level DDoS rules, make sure you do not configure rulebase-ddos rules that have two different application-ddos objects when the traffic destined to one application server can process more than one rule. Essentially, for each protected application server, you have to configure the application-level DDoS rules so that traffic destined for one protected server only processes one application-level DDoS rule.



NOTE: Application-level DDoS rules are terminal, which means that once traffic is processed by one rule, it will not be processed by other rules.

The following configuration options can be committed, but they will not work properly:

source-zone	destination-zone	destination-ip	service	application-ddos	Application Server
source-zone-1	dst-1	any	http	http-appddos1	1.1.1.1:80
source-zone-2	dst-1	any	http	http-appddos2	1.1.1.1:80

- On SRX3400, SRX3600, SRX5600, and SRX5800 devices, application-level DDoS rulebase (rulebase-ddos) does not support port mapping. If you configure an application other than default, and if the application is from either predefined Junos OS applications or a custom application that maps an application service to a nonstandard port, application-level DDoS detection will not work.

When you configure the application setting as default, intrusion detection and prevention (IDP) uses application identification to detect applications running on standard and nonstandard ports; thus, the application-level DDoS detection would work properly.

- On SRX Series and J Series devices, IP actions do not work when you select a timeout value greater than 65,535 in the IDP policy.
- On SRX210, SRX240, and SRX650 devices, the maximum number of IDP sessions supported is 16,000.
- On SRX Series devices, all IDP policy templates are supported except All Attacks. There is a 100-MB policy size limit for integrated mode and a 150-MB policy size limit for dedicated mode, and the current IDP policy templates supported are dynamic, based on the attack signatures being added. Therefore, be aware that supported templates might eventually grow past the policy-size limit.

On SRX Series devices, the following IDP policies are supported:

- DMZ_Services
- DNS_Service
- File_Server

- Getting_Started
- IDP_Default
- Recommended
- Web_Server
- IDP deployed in both active/active and active/passive chassis clusters has the following limitations:
 - No inspection of sessions that fail over or fail back.
 - The IP action table is not synchronized across nodes.
 - The Routing Engine on the secondary node might not be able to reach networks that are reachable only through a Packet Forwarding Engine (PFE).
 - The SSL session ID cache is not synchronized across nodes. If an SSL session reuses a session ID and it happens to be processed on a node other than the one on which the session ID is cached, the SSL session cannot be decrypted and will be bypassed for IDP inspection.
- IDP deployed in active/active chassis clusters has a limitation that for time-binding scope source traffic, if attacks from a source (with more than one destination) have active sessions distributed across nodes, the attack might not be detected because time-binding counting has a local-node-only view. Detecting this sort of attack requires an RTO synchronization of the time-binding state that is not currently supported.

CHAPTER 23

Application-Level Distributed Denial of Service

- IDP Application-Level DDoS Attack Overview on page 523
- IDP Application-Level DDoS Protection Overview on page 523
- Example: Enabling IDP Protection Against Application-Level DDoS Attacks (CLI) on page 528
- Understanding Application-level DDoS Statistic Reporting on page 531
- Example: Configuring Application-level DDoS Statistic Reporting on page 533

IDP Application-Level DDoS Attack Overview

The intent of an application-level DDoS attack is to overwhelm the targeted server, such as a DNS or HTTP servers, so it can not perform its intended services. This is done by making a tremendous amount of application requests from malicious bot clients that often use spoofed IP addresses.

Application-level DDoS attacks are different than traditional Layer 3 and Layer 4 DDoS attacks, such as a SYN flood. From a Layer 3 and Layer 4 perspective, the attack can appear as legitimate transactions. Traditional Layer 3 and Layer 4 DDoS solutions can only rate limit these attacks and begin the application transactions, instead of denying the attacks.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP Policies Overview on page 463
- Understanding IDP Application-Level DDoS Rulebases on page 479
- IDP Application-Level DDoS Protection Overview on page 523
- Example: Enabling IDP Protection Against Application-Level DDoS Attacks (CLI) on page 528

IDP Application-Level DDoS Protection Overview

- Understanding the Application-Level DDoS Module on page 524
- Understanding the Application-Level DDoS Definition on page 525

- Understanding the Application-Level DDoS Rule on page 526
- Understanding Application-Level DDoS IP-Action on page 527
- Understanding Application-Level DDoS Session Action on page 528

Understanding the Application-Level DDoS Module

The application-level distributed denial-of-service (application-level DDoS) IDP module uses application-level metrics to differentiate between normal and malicious application requests. It then identifies the offending source addresses and can drop or deny these requests. Based on user-configured application thresholds, when the client application transactions exceed the defined thresholds, session and ip-actions are applied on traffic from the offending client address. This feature protects servers against DNS and HTTP application-level DDoS attacks.

To identify malicious bot clients, you create a policy with `rulebase-ddos` to monitor specific traffic and define application-level DDoS application metrics and thresholds to monitor that traffic. When the threshold are exceeded, your defined action is taken on the client to protect the application server.

IDP performs multistage analysis from connection monitoring, to protocol analysis, to bot client classification, and maintains the state for each protected server. You can configure `connection-rate-threshold` in the application-level DDoS application definition to monitor connection rate. When connection thresholds rate are exceeded, IDP transitions to protocol analysis for deep content inspection and maintains statistical data on application transactions. The application-level DDoS attacks can be classified into either heavy hitters or random hitters. Heavy hitters perform identical application transactions in a fast and repeated fashion, for example, querying a nonexistent domain name repeatedly. Random hitters perform random application transactions, for example, querying a random domain name, one per each request. You can configure `context value-hit-rate-threshold` to detect heavy hitters and `context hit-rate-threshold` to detect random hitters. If either of the context thresholds are exceeded, IDP transitions to the bot client classification stage, where it tracks the application transactions on a per-client basis based on user-configured time-binding thresholds. A benign client will not perform identical and repeated transactions, whereas malicious bot clients will. Once time-binding thresholds are exceeded, identified bot clients will be blocked with the configured ip-action and session actions.

You can also configure a list of regular expressions under `exclude-context-values` to exempt certain context values from being considered for application-level DDoS processing. This is helpful for requests for well-known resources that can often hit context thresholds, for example, a DNS query for domain name `google.com`.

Protocol analysis stage uses a default interval of 60 seconds for `context hit-rate-threshold` and `value-hit-rate-threshold`. For example, if you configure 10,000 as the `value-hit-rate-threshold`, the context value would be monitored against a 10,000 hits limit in a 60-second interval.

IDP also uses hysteresis for state transitions to avoid thrashing between the states. A default of 20 percent lower limit will be used from the configured connection and context thresholds for falling behind in state. For example, if you configure a context

value-hit-rate-threshold of 10,000, IDP transitions from protocol analysis to bot client classification after 10,000 hits in 60 seconds for identical context values, and falls behind in state only when such hits are smaller than 8000 in 60 seconds.

We recommend configuring time-binding thresholds in the application-ddos definition, because it is critical to differentiate between benign clients and malicious bot clients. However, if you choose not to define time-binding thresholds, IDP will not do bot client classification. In this case, if application transactions exceed context thresholds, the configured ip-action and session actions will be performed. Note that without bot client classification, benign clients might get denied when making a request to the protected server.

IDP maintains application-level DDoS state for the current policy only. For more information, see the policy aging reference in “Understanding Multiple IDP Detector Support” on page 513. Traffic from sessions using older policy will not be inspected for application-level DDoS. If a new policy is loaded, application-level DDoS state for each protected server will be relearned.

Understanding the Application-Level DDoS Definition

You can only configure one application-ddos definition for each protected server. However, you can use the same application-ddos definition in two or more rules with specific destination-address, to-zone, or both to protect two or more servers with the same desired application-ddos thresholds.

Table 55 on page 525 shows the parameters that can be set for application-ddos. For more details, see the *Junos OS CLI Reference* guide.



NOTE: Application-level denial-of-service (application-level DDoS) detection will not work if two rules with different application-level DDoS applications process traffic going to a single destination application server. When setting up application-level DDoS rules, make sure you do not configure rulebase-ddos rules that have two different application-ddos objects while the traffic destined to one application server can process more than one rule. Essentially, for each protected application server, you have to configure application-level DDoS rules so traffic destined for one protected server only hits one application-level DDoS rule.

application-level DDoS rules are terminal, which means that once traffic is processed by one rule, it will not be processed by other rules.

Table 55: Application DDoS Parameters

Parameter	Description
service service-name	The Application Layer service to be monitored, such as DNS or HTTP.

Table 55: Application DDoS Parameters (*continued*)

Parameter	Description
exclude-context-value	Configure a list of common context value patterns that should be excluded from application-level DDoS detection. For example, if you have a webserver that receives a high number of HTTP requests on the home/landing page, you can exclude it from application-level DDoS detection.
connection-rate threshold	The connections-per-second threshold at which to start monitoring the application context values.
context <i>context-name</i>	Name of the application context that the IDP protocol decoder generates while parsing the application protocol from traffic data.
hit-rate-threshold	Number of context hits in tick interval (60 seconds by default) to start bot client classification, if timebinding parameters are configured. If timebinding parameters are not configured, the configured policy actions are taken.
value-hit-rate-threshold	Number of context value hits in tick interval to start bot client classification, if timebinding parameters are configured. If time binding parameters are not configured, the configured policy actions are taken.
max-context-values	The top <i>n</i> context values that should be monitored, reported, or both.
time-binding-period	The time-binding period to determine if a client should be classified as a malicious bot client or not. This setting is used in conjunction with time-binding count to detect an attack if a client request for the same context value exceeds time-binding-count times in time-binding-period seconds.
time-binding-count	The number of context or context value hits from each client over the time binding period to determine if it should be considered a malicious bot client.

Understanding the Application-Level DDoS Rule

You configure one or more application-Level DDoS rules to define the traffic that should be monitored to protect your servers.

Table 56 on page 526 shows the parameters that can be set for application-ddos.

Table 56: application-level DDoS Rule Parameters

Parameter	Description
from-zone	Match source zone.
source-address	Match source address.
to-zone	Match destination zone.

Table 56: application-level DDoS Rule Parameters (*continued*)

Parameter	Description
destination-address	Match destination address.
application	Choose default to select the application service from the application-ddos definition.
application-ddos	Specify the DDoS application.

Understanding Application-Level DDoS IP-Action

You configure ip-action either to drop future sessions from identified bot client addresses for a specified time or to rate-limit future connections.

Table 57 on page 527 shows the available parameters for configuring an application-level DDoS IP action.

Table 57: Application-Level DDoS IP-Action Parameters

Parameter	Description
ip-block	Blocks future connections of any session that matches the IP action.
ip-close	Closes future connections of any client address that matches the IP action by sending an RST packet to the client. If TCP is not used, the connection is dropped silently.
ip-connection-rate-limit	Rate-limits future connections based on a connections per second limit that you set. This parameter can be used to reduce the number of attacks from a client.
ip-notify	Takes no action against matching future connections, but logs the event.
destination-address	Matches traffic based on the destination address of the attack traffic.
service	Matches traffic based on the source address, destination address, destination port, and protocol of the attack traffic. This is the default.
source-address	Matches traffic based on the source address of the attack traffic.
source-zone	Matches traffic based on the source zone of the attack traffic.
zone-service	Matches traffic based on the source zone, destination address, destination port, and protocol of the attack traffic.
log	Logs the information about the IP action against the traffic that matches a rule.
timeout	Specifies the number of seconds that you want the IP action to remain in effect after a traffic match.

Understanding Application-Level DDoS Session Action

Session action determines what action should be performed on the identified bot client.

Table 58 on page 528 shows the parameters that can be set for action.

Table 58: application-level DDoS Action Parameters

Parameter	Description
close-server	Closes the connection and sends an RST packet to the server but not to the client.
drop-connection	Drops all packets associated with the connection, preventing traffic for the connection from reaching its destination. Use this action to drop connections for traffic that is not prone to spoofing.
drop-packet	Drops a matching packet before it can reach its destination but does not close the connection. Use this action to drop packets for attacks in traffic that is prone to spoofing, such as UDP traffic. Dropping a connection for such traffic could result in a denial of service that prevents you from receiving traffic from a legitimate source-IP address.
no-action	No action is taken. Use this action when you want to generate logs for only some traffic.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Application-Level DDoS Rulebases on page 479
 - IDP Application-Level DDoS Attack Overview on page 523
 - Example: Enabling IDP Protection Against Application-Level DDoS Attacks (CLI) on page 528

Example: Enabling IDP Protection Against Application-Level DDoS Attacks (CLI)

This example describes how you can use the application-level distributed denial-of-service (application-level DDoS) module to protect a DNS server.

When setting up application-level DDoS protection for a DNS server, you will first observe the average load of DNS requests. If the DNS server is expected to handle a normal load of 1000 requests per second, choose 20 percent in excess of the normal load (1200 requests per second) as the connection-rate-threshold. This is essentially 60,000 transactions in 60 seconds, so choose 20percent in excess of this load as context hit-rate-threshold (72,000). You can choose context value-hit-rate-threshold based on the maximum load of requests for the same domain name being queried for nonexempt context dns-type-cname values. For example, if it is impractical for DNS to receive queries for domain xyz.com in excess of 2000 times in 60 seconds, context value-hit-rate-threshold should be set to 20percent more than that value, which would be 2400 times in 60 seconds. For monitoring and reporting, you will optionally set max-context-values to 100, so at the maximum, the most active 100 DNS query requests

will be monitored and reported. If a client is in this range, it is mostly likely a malicious bot client. Once bot clients are identified, you can configure ip-action as ip-block with timeout as 600 seconds (the bot client gets access denied for 1 hour) and session action is set as drop-packet.

In the example, IDP starts deep protocol analysis when the number of connections per second exceeds 1200, and will start bot client classification if either the total number of queries for context dns-type-name exceeds 72,000 or if requests for the same query value exceeds 2400.



NOTE: When an application-level DDoS attack occurs on the application server, it will have much higher transaction rates than it does under normal or even peak load. With this in mind, it is best to set higher thresholds than the normal peak of the application server so it does not trigger unnecessary client classification processing. This will improve the over-all performance of the Juniper device because the application-level DDoS module will not start client classification until the server has actually reached abnormal transaction rates.

For detailed information about the following commands, see the *Junos OS CLI Reference*.

To enable protection against application-level DDoS for a DNS server, use the CLI configuration editor.

1. Access the IDP **security** configuration hierarchy.

```
[edit]
user@host# edit security idp
```

2. Configure application-ddos to define the type of traffic, the protocol context that will be monitored, and thresholds that will be used to trigger an action. In this example, DNS traffic and the protocol context dns-type-name will be monitored.

```
[edit security idp]
user@host# set application-ddos dns-server-1
[edit security idp]
user@host# set application-ddos dns-server service dns
[edit security idp]
user@host# set application-ddos dns-server-1 connection-rate-threshold 1200
[edit security idp]
user@host# set application-ddos dns-server-1 context dns-type-name
    hit-rate-threshold 72000
[edit security idp]
user@host# set application-ddos dns-server-1 context dns-type-name
    value-hit-rate-threshold 2400
[edit security idp]
user@host# set application-ddos dns-server-1 context dns-type-name
    max-context-values 100
[edit security idp]
user@host# set application-ddos dns-server-1 context dns-type-name
    time-binding-count 10
[edit security idp]
user@host# set application-ddos dns-server-1 context dns-type-name
    time-binding-period 30
```



NOTE: You can continue to set other protocol contexts and thresholds that you would like to monitor.

3. (Optional) Set context values that will be exempt from monitoring.

```
[edit security idp]
user@host# set application-ddos dns-server-1 context dns-type-name
exclude-context-values .*google.com
[edit security idp]
user@host# set application-ddos dns-server-1 context dns-type-name
exclude-context-values .*yahoo.com
```

4. Set the IDP policy rule for rulebase-ddos to define the source and destination of traffic that will be monitored.



NOTE: You can only define one DDoS application per application-level DDoS rule. Create additional rules to monitor multiple DDoS applications.

Each application-level DDoS rule is a terminal rule, meaning that only one matching rule is considered for incoming traffic matching.

```
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
match source-address any
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
match to-zone any
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
match destination-address any
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
match application default
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
match application-ddos dns-server-1
[edit security idp]
```

5. Define the action to be taken when application-level DDoS attack traffic is detected.

```
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
then action drop-packet
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
then ip-action ip-block
[edit security idp]
user@host# set idp-policy AppDDoS-policy-1 rulebase-ddos rule AppDDoS-rule1
then ip-action timeout 600
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Application-Level DDoS Rulebases on page 479

- IDP Application-Level DDoS Attack Overview on page 523
- IDP Application-Level DDoS Protection Overview on page 523

Understanding Application-level DDoS Statistic Reporting

To successfully mitigate application-level distributed denial-of-service (DDoS) attacks on your network environment, you need to set the appropriate rule thresholds. To identify the appropriate thresholds, you need to analyze network statistical data. With application-level DDoS statistic reporting, you can collect application information on connection, context and rates, and data records from application requests destined for your protected servers. With this information, you can determine trends to help you create more efficient rules for your environment.

Following are the main features of statistic reporting:

- Application-level DDoS reporting on application request data.
- Statistics collection of application connection rates and context rates on a periodic basis that you define. The default snapshot interval is once every 1 minute and the range is 1 through 60 minutes.
- Report files are written to the Routing Engine (RE) data storage device for extensive storage space.
- Automatic file compression of statistical report files when file size reaches 10 MB.



NOTE: Statistic reports are saved on the Routing Engine (RE) data storage device in the `/var/log/addos` directory. There must be at least 2 GB of free space to allow report logging.

The IDP module polls for application-level DDoS records and takes a snapshot of current activity at intervals that you define. Each statistical record collected represents an application request data entry (context value) up to 4 KB. Information collected includes the server IP address, zone, connection, and context rates, protocol, and Layer 7 service and context values. The `max-context-values` setting determines how many records should be collected per application context.

The filenaming convention for reports stored in `/var/log/addos` comprises the prefix `addos-stats` along with the record creation timestamp in the format `YYYYMMDDHHMMSS` (year/month/day/hour/minute/seconds). For example: `addos-stats-20100501091500`, is May 1, 2010 at 9:15 AM.

The report files are in comma-separated value (.csv) format and should be copied off the device to be analyzed in a program that can read .csv files, such as Excel. See Table 59 on page 532 for descriptions of each field in the application-level DDoS statistic record.

Table 59: Application-Level DDoS Statistic Record Fields

Field	Description
time	Time the event occurred.
record-type	Type of record that is created. Type app-record is supported.
record-data	Identifies the type of data collected (addos-http-url or addos-dns).
destination-ip	Destination IP of the application request.
ddos-app-name	Name of the configured application object defined in the application-level DDoS rule.
conn/sec	Connection attempts per second by the application.
context-name	Context name in the application header.
context-hits/tick	Number of context hits per tick interval. The default tick interval is 60 seconds.
context-value-hits/tick	Number of context value hits per tick interval. The default tick interval is 60 seconds.
context-value	Application context name. The context-value is reported both in hexadecimal and ASCII formats and is no larger than 4 K.

The following output shows an application-level DDoS statistic record.

```
2010:01:16:04:23:53,app-record,my-http,5.0.0.1,trust,6,http-url-parsed,1234/60sec,1234/60sec,ascii:/abc.html
hex:2f6162632e68746d6c
2010:01:16:04:23:53,app-record,my-http,5.0.0.1,trust,6,http-url-parsed,932791/60sec,932791/60sec,ascii:/index.html
hex:2f696e6465782e68746d6c
```

The following screen shot shows a formatted application-level DDoS statistic report.

time	record-type	record-data	destination-ip	ddos-app-name	conn/sec	context-name	context-hits/tick	context-value-hits/tick	context-value
"2010:01:22:20:56:34"	app-record	addos-http-url	61.0.3.43	bps-servers	24	http-header-user-agent	199/60sec	199/60sec	ascii:Mozilla/4.0 (compatible; MSIE 6.0;
"2010:01:22:20:56:34"	app-record	addos-http-url	61.0.3.7	bps-servers	31	http-header-user-agent	196/60sec	196/60sec	ascii:Mozilla/4.0 (compatible; MSIE 6.0;
"2010:01:22:20:56:34"	app-record	addos-dns	71.2.0.72	test-servers	16	dns-type-name	330/60sec	11/60sec	ascii...host28 hex:0001686f73743238"
"2010:01:22:20:56:34"	app-record	addos-dns	71.2.0.94	test-servers	23	dns-type-name	529/60sec	11/60sec	ascii...host28 hex:0001686f73743238"



NOTE: To clear out statistics files that are no longer needed, you run the operational command `request system storage cleanup`.

You can use the statistical data you collect to analyze application-level DDoS activity and identify the types and rates of application activity hitting your server. Typically, you will initially set your rules to have low thresholds with no action; then, once you profile your environment by analyzing the collected statistics, you can protect your servers by setting appropriate limits and configuring effective actions for attacks.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Configuring Application-level DDoS Statistic Reporting on page 533

- IDP Application-Level DDoS Attack Overview on page 523
- Understanding IDP Policy Rules on page 470
- Understanding IDP Application-Level DDoS Rulebases on page 479

Example: Configuring Application-level DDoS Statistic Reporting

The configuration instructions in this topic describe how to enable application-level DDoS statistic reporting.

In this example, you will enable statistics reporting with a time interval of 5 minutes. With this setting, at 5 minute intervals the application-level DDoS module will take a snapshot of current activity and will place a report file in the `/var/log/addos` folder in comma separated (CSV) format.

To enable application-level DDoS statistic reporting with a time interval of 5 minutes:

```
user@host# set security idp sensor-configuration application-ddos statistics interval 5
```

To disable application-level DDoS statistic reporting:

```
user@host# delete security idp sensor-configuration application-ddos statistics
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Application-level DDoS Statistic Reporting on page 531
 - IDP Application-Level DDoS Attack Overview on page 523
 - Understanding IDP Policy Rules on page 470
 - Understanding IDP Application-Level DDoS Rulebases on page 479

IDP Signature Database

- Understanding the IDP Signature Database on page 535
- Example: Adding a Detector Sensor Configuration (J-Web) on page 536
- Predefined IDP Policy Templates on page 537
- IDP Signature Databases on page 539
- Verifying the Signature Database on page 545

Understanding the IDP Signature Database

The signature database is one of the major components of Intrusion Detection and Prevention (IDP). It contains definitions of different objects—such as attack objects, application signatures objects, and service objects—that are used in defining IDP policy rules. As a response to new vulnerabilities, Juniper Networks periodically provides a file containing attack database updates on the Juniper website. You can download this file to protect your network from new threats.

The IDP signature database is stored on the IDP enabled device and contains definitions of predefined attack objects and groups. These attack objects and groups are designed to detect known attack patterns and protocol anomalies within the network traffic. You can configure attack objects and groups as match conditions in IDP policy rules.



NOTE: You must install the IDP signature-database-update license key on your device for downloading and installing daily signature database updates provided by Juniper Networks. The IDP signature license key does not provide grace period support. For license details, see the *Junos OS Administration Guide for Security Devices*.

You can perform the following tasks to manage the IDP signature database:

- Update the signature database—Download the attack database updates available on the Juniper Networks website. New attacks are discovered daily, so it is important to keep your signature database up to date.
- Verify the signature database version—Each signature database has a different version number with the latest database having the highest number. You can use the CLI to display the signature database version number.

- Update the protocol detector engine—You can download the protocol detector engine updates along with downloading the signature database. The IDP protocol detector contains Application Layer protocol decoders. The detector is coupled with the IDP policy and is updated together. It is always needed at policy update time, even if there is no change in the detector.
- Schedule signature database updates—You can configure the IDP-enabled device to automatically update the signature database after a set interval.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP Policies Overview on page 463
- Understanding IDP Policy Rulebases on page 476
- Understanding IDP Policy Rules on page 470
- Example: Defining Rules for an IDP IPS Rulebase on page 481
- Understanding Predefined IDP Policy Templates on page 537
- Example: Updating the IDP Signature Database Manually (CLI) on page 543
- Example: Updating the Signature Database Automatically (CLI) on page 545

Example: Adding a Detector Sensor Configuration (J-Web)

In this example, you add a detector sensor configuration for File Transfer Protocol (FTP) with Tunable Value 1000, which can be further tuned using the Basic and Advanced configuration tabs on the IDP Sensor configuration page.

To add a detector sensor configuration:

1. Select **Configure>Security>IDP>Sensor**.
2. Select the **Detector** tab.
3. Click **Add**.
4. In the Protocol list, select **FTP**.
5. In the Tunable Name list, select **sc_ftp_flags**.
6. In the Tunable Value box, type **1000**.
7. Click **OK** to save the configuration.



NOTE: For more information about how to configure this feature using the J-Web Configure menu, navigate to the **Configure>Security>IDP>Sensor** page in the J-Web user interface and click **Help**.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP Policies Overview on page 463

- Understanding the IDP Signature Database on page 535
- Understanding IDP Application Identification on page 549
- Example: Configuring IDP Policies for Application Identification (CLI) on page 551

Predefined IDP Policy Templates

- Understanding Predefined IDP Policy Templates on page 537
- Downloading and Using Predefined IDP Policy Templates (CLI Procedure) on page 538

Understanding Predefined IDP Policy Templates

Juniper Networks provides predefined policy templates that you can use as a starting point for creating your own policies. Each template is set of rules of a specific rulebase type that you can copy and then update according to your requirements. These templates are available in the **templates.xml** file on a secured Juniper Networks website. To start using a template, you run a command from the CLI to download and copy this file to a **/var/db/scripts/commit** directory.

Each policy template contains rules that use the default actions associated with the attack objects. You should customize these templates to work on your network by selecting your own source and destination addresses and choosing IDP actions that reflect your security needs.

Table 60 on page 537 summarizes the predefined IDP policy templates provided by Juniper Networks.

Table 60: Predefined IDP Policy Templates

Template Name	Description
All With Logging	Includes all Attack Objects and enables packet logging for all rules.
All Without Logging	Includes all Attack Objects but does not enable packet logging.
DMZ Services	Protects a typical demilitarized zone (DMZ) environment.
DNS Server	Protects Domain Name System (DNS) services.
File Server	Protects file sharing services, such as Network File System (NFS), FTP, and others.
Getting Started	Contains very open rules. Useful in controlled lab environments, but should not be deployed on heavy traffic live networks.
IDP Default	Contains a good blend of security and performance.
Recommended	Contains only the attack objects tagged as <i>recommended</i> by Juniper Networks. All rules have their Actions column set to take the recommended action for each attack object.
Web Server	Protects HTTP servers from remote attacks.

To use predefined policy templates:

1. Download the policy templates from the Juniper Networks website.
2. Install the policy templates.
3. Enable the **templates.xml** script file. Commit scripts in the **/var/db/scripts/commit** directory are ignored if they are not enabled.
4. Choose a policy template that is appropriate for you and customize it if you need to.
5. Activate the policy that you want to run on the system. Activating the policy might take a few minutes. Even after a commit complete message is displayed in the CLI, the system might continue to compile and push the policy to the dataplane.



NOTE: Occasionally, the compilation process might fail for a policy. In this case, the active policy showing in your configuration might not match the actual policy running on your device. Run the **show security idp status** command to verify the running policy. Additionally, you can view the IDP log files to verify the policy load and compilation status (see “Verifying the Signature Database” on page 545).

6. Delete or deactivate the commit script file. By deleting the commit script file, you avoid the risk of overwriting modifications to the template when you commit the configuration. Deactivating the statement adds an inactive tag to the statement, effectively commenting out the statement from the configuration. Statements marked inactive do not take effect when you issue the **commit** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP Policies Overview on page 463
 - Understanding IDP Policy Rulebases on page 476
 - Understanding IDP Policy Rules on page 470
 - Downloading and Using Predefined IDP Policy Templates (CLI Procedure) on page 538

Downloading and Using Predefined IDP Policy Templates (CLI Procedure)

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

To download and use a predefined policy template:

1. Download the script file **templates.xml** to the **/var/db/idpd/sec-download/sub-download** directory. This script file contains predefined IDP policy templates.

```
user@host> request security idp security-package download policy-templates
```
2. Copy the **templates.xml** file to the **/var/db/scripts/commit** directory and rename it to **templates.xsl**.

```
user@host> request security idp security-package install policy-templates
```

3. Enable the **templates.xml** scripts file. At commit time, the Junos OS management process (mgd) looks in the **/var/db/scripts/commit** directory for scripts and runs the script against the candidate configuration database to ensure the configuration conforms to the rules dictated by the scripts.

```
user@host# set system scripts commit file templates.xml
```

4. Commit the configuration. Committing the configuration saves the downloaded templates to the Junos OS configuration database and makes them available in the CLI at the **[edit security idp idp-policy]** hierarchy level.
5. Display the list of downloaded templates.

```
user@host# set security idp active-policy ?
```

```
Possible completions:
<active policy> Set active policy
All_With_Logging
All_Without_Logging
DMZ_Services
DNS_Service
File_Server
Getting_Started
IDP_Default
Recommended
Web_Server
```

6. Activate the predefined policy. The following statement specifies the *Recommended* predefined IDP policy as the active policy:

```
user@host# set security idp active-policy Recommended
```

7. Delete or deactivate the commit script file. By deleting the commit script file, you avoid the risk of overwriting modifications to the template when you commit the configuration. Run one of the following commands:

```
user@host# delete system scripts commit file templates.xml
user@host# deactivate system scripts commit file templates.xml
```

8. If you are finished configuring the device, commit the configuration.
9. You can verify the configuration by using the **show security idp status** command. For more information, see the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Predefined IDP Policy Templates on page 537
 - Example: Defining Rules for an IDP IPS Rulebase on page 481
 - Example: Defining Rules for an IDP Exempt Rulebase on page 485

IDP Signature Databases

- Understanding Predefined IDP Attack Objects and Object Groups on page 540
- Understanding the IDP Signature Database Version on page 541

- Updating the IDP Signature Database Overview on page 542
- Updating the IDP Signature Database Manually Overview on page 543
- Example: Updating the IDP Signature Database Manually (CLI) on page 543
- Example: Updating the Signature Database Automatically (CLI) on page 545

Understanding Predefined IDP Attack Objects and Object Groups

The security package for Intrusion Detection and Prevention (IDP) contains a database of predefined IDP attack objects and IDP attack object groups that you can use in IDP policies to match traffic against known and unknown attacks. Juniper Networks updates the predefined attack objects and groups on a regular basis with newly discovered attack patterns.

Updates to the attack object database can include:

- New descriptions or severities for existing attack objects
- New attack objects
- Deletion of obsolete attack objects

This topic includes the following sections:

- Predefined Attack Objects on page 540
- Predefined Attack Object Groups on page 540

Predefined Attack Objects

Predefined attack objects are listed in an alphabetical order. These attack objects have unique names that help you identify the attack. The first part of the name indicates the group to which the attack object belongs. For example:

- **FTP:USER:ROOT**—Belongs to the **FTP:USER** group. It detects attempts to log in to an FTP server using the **root** account.
- **HTTP:HOTMAIL:FILE-UPLOAD**—Belongs to the **HTTP:HOTMAIL** group. It detects files attached to e-mails sent via the Web-based e-mail service **Hotmail**.

Predefined Attack Object Groups

The predefined attack groups list displays the attack objects in the categories described below. A set of recommended attack objects that Juniper Networks considers to be serious threats are also available in this list. The recommended attack objects are organized into the following categories:

Table 61: Predefined Attack Object Groups

Attack Object Group	Description
Attack Type	Groups attack objects by type (anomaly or signature). Within each type, attack objects are grouped by severity.

Table 61: Predefined Attack Object Groups (*continued*)

Attack Object Group	Description
Category	Groups attack objects by predefined categories. Within each category, attack objects are grouped by severity.
Operating System	Groups attack objects by the operating system to which they apply: BSD, Linux, Solaris, or Windows. Within each operating system, attack objects are grouped by services and severity.
Severity	Groups attack objects by the severity assigned to the attack. IDP has five severity levels: Critical, Major, Minor, Warning, Info. Within each severity, attack objects are grouped by category.
Web Services	Groups attack objects by common Web services. These services are grouped by severity levels—Warning, Critical, Major, Minor, Info.
Miscellaneous	Groups attack objects by performance level. Attack objects affecting IDP performance over a certain level are grouped under this category.
Response	Groups attack objects in traffic flowing in the server to client direction.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding the IDP Signature Database on page 535
 - Updating the IDP Signature Database Manually Overview on page 543
 - IDP Policies Overview on page 463
 - Understanding IDP Policy Rulebases on page 476
 - Understanding IDP Policy Rules on page 470
 - Example: Defining Rules for an IDP IPS Rulebase on page 481
 - Example: Defining Rules for an IDP Exempt Rulebase on page 485

Understanding the IDP Signature Database Version

New attack objects are added to the signature database server frequently; downloading these updates and installing them on your managed devices regularly ensures that your network is effectively protected against the latest threats. As new attack objects are added to the signature database server, the version number of the database is updated with the latest database version number. Each signature database has a different version number with the latest database having the highest number.

When updating the signature database, the signature database update client connects to the Juniper Networks website and obtains the update using an HTTPS connection. This update—difference between the existing signature database and latest signature database—is calculated based on the version number that is assigned to each signature database. After you download the updates, the updated information is merged with the

existing signature database and the version number is set to that of the latest signature database.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Predefined IDP Attack Objects and Object Groups on page 540
 - Understanding the IDP Signature Database on page 535
 - IDP Policies Overview on page 463
 - Understanding IDP Policy Rules on page 470
 - Example: Updating the IDP Signature Database Manually (CLI) on page 543
 - Example: Updating the Signature Database Automatically (CLI) on page 545

Updating the IDP Signature Database Overview

Juniper Networks regularly updates the predefined attack database and makes it available on the Juniper Networks website. This database includes attack object groups that you can use in Intrusion Detection and Prevention (IDP) policies to match traffic against known attacks. Although you cannot create, edit, or delete predefined attack objects, you can use the CLI to update the list of attack objects that you can use in IDP policies.

To update the signature database, you download a security package from the Juniper Networks website. The security package consists of the following IDP components:

- Attack objects
- Attack object groups
- Application objects
- Updates to the IDP Detector Engine
- IDP Policy templates (Policy templates are downloaded independently. See “Understanding Predefined IDP Policy Templates” on page 537.)

By default, when you download the security package, you download the following components into a Staging folder in your device: the latest version of the complete attack object groups table, application objects table, and the updates to the IDP Detector Engine. Because the attack objects table is typically of a large size, by default the system downloads only updates to the attack objects table. However, you can download the complete attack objects table by using the **full-update** configuration option.

After downloading the security package, you must install the package to update the security database with the newly downloaded updates from the Staging folder in your device.

After installing a security package, when you commit the configuration, all policies are checked for their syntax (not only the active policy). This checking is the same as a commit check. If an attack configured in any of the existing policies is removed from the new signature database that you download, the commit check fails. When you update the IDP signature database, attacks configured in policies are not updated automatically. For example, suppose you configure a policy to include an attack **FTP:USER:ROOT** that

is available in the signature database version 1200 on your system. Then, you download signature database version 1201, which no longer includes the attack **FTP:USER:ROOT**. Because an attack configured in your policy is missing from the newly downloaded database, the commit check in the CLI fails. To successfully commit your configuration, you must remove the attack (**FTP:USER:ROOT**) from your policy configuration.



CAUTION: IDP signature updates might fail if a new IDP policy load fails for any reason. When a new IDP policy load fails, the last known good IDP policy is loaded. Once the issue with the new policy load is resolved, and the new valid policy is active, signature updates will work properly.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Predefined IDP Attack Objects and Object Groups on page 540
- Understanding the IDP Signature Database on page 535
- IDP Policies Overview on page 463
- Understanding IDP Policy Rules on page 470
- Example: Updating the IDP Signature Database Manually (CLI) on page 543
- Example: Updating the Signature Database Automatically (CLI) on page 545

Updating the IDP Signature Database Manually Overview

Juniper Networks regularly updates the predefined attack database and makes it available on the Juniper Networks website. This database includes attack object groups that you can use in Intrusion Detection and Prevention (IDP) policies to match traffic against known attacks. Although you cannot create, edit, or delete predefined attack objects, you can use the CLI to update the list of attack objects that you can use in IDP policies. After downloading the security package, you must install the package to update the security database with the newly downloaded updates from the Staging folder in your device.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP Policies Overview on page 463
- Understanding Predefined IDP Attack Objects and Object Groups on page 540
- Understanding the IDP Signature Database on page 535
- Understanding the IDP Signature Database Version on page 541
- Example: Updating the IDP Signature Database Manually (CLI) on page 543
- Example: Updating the Signature Database Automatically (CLI) on page 545

Example: Updating the IDP Signature Database Manually (CLI)

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

The configuration instructions in this topic describe how to download the security package with the complete table of attack objects and attack object groups, create a policy, and specify the new policy as the active policy. This example then describes how to download only the updates that Juniper Networks has recently uploaded and then update the attack database, running policy, and detector with these new updates.

To manually download and update the signature database:

1. Download the security package. The security package includes the detector and the latest attack objects and groups.

```
user@host> request security idp security-package download full-update
```
2. Update the attack database, the active policy, and the detector with the new package.

```
user@host> request security idp security-package install
```
3. Check the attack database update status with the following command. The command output displays information about the downloaded and installed versions of attack database versions.

```
user@host> request security idp security-package install status
```
4. Commit the configuration.
5. After committing the configuration, the attack objects and groups are available in the CLI under the **predefined-attack-groups** and **predefined-attacks** configuration statements at the **[edit security idp idp-policy]** hierarchy level.
6. Associate attack objects or attack object groups with the policy. The following statement associates the recommended attack object group **Response_Critical-TELNET** with **policy1**:

```
user@host# set security idp idp-policy policy1 rulebase-ips rule rule1 match attacks predefined-attack-groups "Response_Critical - TELNET"
```
7. Activate the policy. The following statement makes **policy1** the active policy on the device:

```
user@host# set security idp active-policy policy1
```
8. Commit the configuration.
9. After a week, if you want to download only the updates that Juniper Networks has recently uploaded, use the following command:

```
user@host> request security idp security-package download
```
10. Update the attack database, active policy, detector with the new changes:

```
user@host> request security idp security-package install
```
11. If you are finished configuring the device, commit the configuration.
12. From configuration mode in the CLI, enter the **show security idp** command to verify the configuration. For more information, see the *Junos OS CLI Reference*.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Updating the IDP Signature Database Manually Overview on page 543

- Example: Updating the Signature Database Automatically (CLI) on page 545
- Understanding the IDP Signature Database on page 535

Example: Updating the Signature Database Automatically (CLI)

Juniper Networks regularly updates the predefined attack database and makes it available on the Juniper Networks website. This database includes attack object groups that you can use in IDP policies to match traffic against known attacks. You can configure your device to download the signature database updates automatically at a specified interval.

The configuration instructions in this topic describe how to download the security package with the complete table of attack objects and attack object groups every 48 hours starting at 11:59 pm on December 10.

To download and update predefined attack objects:

1. Specify the URL for the security package. The security package includes the detector and the latest attack objects and groups. The following statement specifies **`https://services.netscreen.com/cgi-bin/index.cgi`** as the URL for downloading signature database updates:

```
user@host# set security idp security-package url  
https://services.netscreen.com/cgi-bin/index.cgi
```
2. Specify the time and interval for download. The following statement sets the interval as **48** hours and the start time as **11:59** pm on **December 10**:

```
user@host# set security idp security-package automatic interval 48 start-time  
12-10.23:59
```
3. Enable an automatic download and update of the security package.

```
user@host# set security idp security-package automatic enable
```
4. If you are finished configuring the device, commit the configuration.
5. From configuration mode in the CLI, enter the **`show security idp`** command to verify the configuration. For more information, see the *Junos OS CLI Reference*.

Related Topics • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Updating the IDP Signature Database Manually Overview on page 543
- Example: Updating the Signature Database Automatically (CLI) on page 545
- Understanding the IDP Signature Database on page 535

Verifying the Signature Database

- Verifying the IDP Policy Compilation and Load Status on page 546
- Verifying the IDP Signature Database Version on page 547

Verifying the IDP Policy Compilation and Load Status

Purpose Display the IDP log files to verify the IDP policy load and compilation status. When activating an IDP policy, you can view the IDP logs and verify if the policy is loaded and compiled successfully.

Action To track the load and compilation progress of an IDP policy, configure either one or both of the following in the CLI:

- You can configure a log file, which will be located in `/var/log/`, and set trace option flags to record these operations:

```
user@host# set security idp traceoptions file idpd
user@host# set security idp traceoptions flag all
```

- You can configure your device to log system log messages to a file in the `/var/log` directory:

```
user@host# set system syslog file messages any any
```

After committing the configuration in the CLI, enter either of the following commands from the shell prompt in the UNIX-level shell:

Sample Output

```
user@host> start shell
user@host% tail -f /var/log/idpd
Jun  9 18:15:40 logmsg <valid license found for feature 20>
Jun  9 18:15:40 IDP feature license status: Valid license installed.
Jun  9 18:15:40 idpd commit start...
Jun  9 18:15:40 Entering enable processing.
Jun  9 18:15:40 Enable value (default)
Jun  9 18:15:40 IDP processing default.
...
Jun  9 18:15:40 Apply policy configuration, policy ops bitmask = 45
Jun  9 18:15:40 Starting policy (idpengine) compile...
Jun  9 18:16:10 policy compilation memory estimate: 57126048
Jun  9 18:16:10 ...Passed (Shows that the policy compilation is
successful)Jun  9 18:16:10 Starting policy package...
Jun  9 18:16:12 ...Policy Packaging Passed
Jun  9 18:16:12 Starting policy load...
Jun  9 18:16:12 Loading policy(/var/db/idpd/bins/idpengine.bin.gz.v +
/var/db/idpd/sec-repository/libidp-detector.so.gz.v +
/var/db/idpd/bins/compiled_ai.bin)...
Jun  9 18:16:12 idpd_dev_add_ipc_connection called..
...
Jun  9 18:16:20 Reading sensor config...
Jun  9 18:16:20 sensor/idp node does not exist, apply defaults
Jun  9 18:16:20 idpd_dev_add_ipc_connection called...
Jun  9 18:16:20 idpd_dev_add_ipc_connection: done.
...
Jun  9 18:16:20 sensor conf successful
Jun  9 18:16:20
...idpd commit end

Jun  9 18:16:20 Returning from commit mode, status = 0. (Shows the policy load
is successful)
```

Sample Output

```
user@host> start shell
user@host% tail -f /var/log/messages
```

```

Jun 24 17:34:38 turtlebert mgd[4786]: UI_COMMIT_PROGRESS: Commit operation in
progress: activating '/var/run/db/juniper.data'
Jun 24 17:34:38 turtlebert mgd[4786]: UI_COMMIT_PROGRESS: Commit operation in
progress: notifying daemons of new configuration
Jun 24 17:34:38 turtlebert mgd[4786]: UI_COMMIT_PROGRESS: Commit operation in
progress: notifying idpd(62)
Jun 24 17:34:38 turtlebert mgd[4786]: UI_COMMIT_PROGRESS: Commit operation in
progress: signaling 'IDP policy daemon', pid 4699, signal 1, status 0 with
notification errors enabled
...
Jun 24 17:34:45 turtlebert idpd[4699]: IDP_POLICY_LOAD_SUCCEEDED: IDP
policy[/var/db/idpd/bins/test.bin.gz.v] and
detector[/var/db/idpd/sec-repository/libidp-detector.so.gz.v] loaded successfully.
IDPD Trace file:
...
Jun 24 12:10:27 idpd_policy_load: idp policy pre-install succeeded
Jun 24 12:10:27 idpd_comm_server_get_event:478: evGetNext got event.
Jun 24 12:10:27 idpd_comm_server_get_event:486: evDispatch OK
...
Jun 24 12:10:27 idpd_policy_load: idp policy install succeeded
Jun 24 12:10:27 idpd_comm_server_get_event:486: evDispatch OK
...
Jun 24 12:10:27 idpd_policy_load: idp policy post-install succeeded
Jun 24 12:10:28 Reading sensor config...
Jun 24 12:10:28 sensor/idp node does not exist, apply defaults

Jun 24 12:10:28 sensor conf successful
Jun 24 12:10:28

...idpd commit end
Jun 24 12:10:28 Returning from commit mode, status = 0.

```

Meaning Displays log messages showing the procedures that run in the background after you commit the **set security idp active-policy** command. This sample output shows that the policy compilation, sensor configuration, and policy load are successful.

Verifying the IDP Signature Database Version

Purpose Display the signature database version.

Action From the operational mode in the CLI, enter **show security idp security-package-version**.

Sample Output

```

user@host> show security idp security-package-version
Attack database version:31(Wed Apr 16 15:53:46 2008)
Detector version :9.1.140080400
Policy template version :N/A

```

Meaning The output displays the version numbers for the signature database, protocol detector, and the policy template on the IDP-enabled device. Verify the following information:

- **Attack database version**—On April 16, 2008, the version of the signature database active on the device is **31**.
- **Detector version**—Displays the version number of the IDP protocol detector currently running on the device.

- **Policy template version**—Displays the version of the policy template that is installed in the `/var/db/scripts/commit` directory when you run the **request security idp security-package install policy-templates** configuration statement in the CLI.

For a complete description of **show security idp security-package-version** output, see the *Junos OS CLI Reference*.

IDP Application Identification

- Understanding IDP Application Identification on page 549
- Understanding IDP Service and Application Bindings by Attack Objects on page 550
- Example: Configuring IDP Policies for Application Identification (CLI) on page 551
- Disabling Application Identification for an IDP Policy (CLI Procedure) on page 552
- IDP Application Identification for Nested Applications on page 553
- IDP Application System Cache on page 554
- IDP Memory and Session Limits on page 557
- Verifying IDP Counters for Application Identification Processes on page 559

Understanding IDP Application Identification

Juniper Networks provides predefined application signatures that detect Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) applications running on nonstandard ports. Identifying these applications allows Intrusion Detection and Prevention (IDP) to apply appropriate attack objects to applications running on nonstandard ports. It also improves performance by narrowing the scope of attack signatures for applications without decoders.

The IDP sensor monitors the network and detects suspicious and anomalous network traffic based on specific rules defined in IDP rulebases. It applies attack objects to traffic based on protocols or applications. Application signatures enable the sensor to identify known and unknown applications running on nonstandard ports and to apply the correct attack objects.

Application signatures are available as part of the security package provided by Juniper Networks. You download predefined application signatures along with the security package updates. You cannot create application signatures. For information on downloading the security package, see “Updating the IDP Signature Database Manually Overview” on page 543.

The application signatures identify an application by matching patterns in the first packet of a session. The IDP sensor matches patterns for both client-to-server and server-to-client sessions.

Application identification is enabled by default and is automatically turned on when you configure the default application in the IDP policy. However, when you specify an

application in the policy rule, application identification is disabled and attack objects are applied based on the specified application. This specific application configuration overwrites the automatic identification process. For instructions on specifying applications in policy rules, see “Example: Configuring IDP Applications and Services” on page 494.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding the IDP Signature Database on page 535
 - IDP Policies Overview on page 463
 - Understanding IDP Service and Application Bindings by Attack Objects on page 550

Understanding IDP Service and Application Bindings by Attack Objects

Attack objects can bind to applications and services in different ways:

- Attack objects can bind to an application implicitly and not have a service definition. They bind to an application based on the name of a context or anomaly.
- Attack objects can bind to a service using a service name.
- Attack objects can bind to a service using TCP or UDP ports, ICMP types or codes or RPC program numbers.

Whether the specified application or service binding applies or not depends on the complete attack object definition as well as the IDP policy configuration:

- If you specify an application in an attack object definition, the service field is ignored. The attack object binds to the application instead of the specified service. However, if you specify a service and no application in the attack object definition, the attack object binds to the service. Table 62 on page 550 summarizes the behavior of application and service bindings with application identification.

Table 62: Applications and Services with Application Identification

Attack Object Fields	Binding Behavior	Application Identification
:application (http) :service (smtp)	<ul style="list-style-type: none"> • Binds to the application HTTP. • The service field is ignored. 	Enabled
:service (http)	Binds to the application HTTP .	Enabled
:service (tcp/80)	Binds to TCP port 80.	Disabled

For example, in the following attack object definition, the attack object binds to the application **HTTP**, the application identification is enabled, and the service field **SMTP** is ignored.

```

: ("http-test"
 :application ("http")
 :service ("smtp"))

```

```

:rectype (signature)
:signature (
:pattern (".*TERM=xterm; export TERM=xterm; exec bash - i\x0a\x.*")
:type (stream)
)
:type (attack-ip)
)

```

- If an attack object is based on service specific contexts (for example, **http-url**) and anomalies (for example, **tftp_file_name_too_long**), both application and service fields are ignored. Service contexts and anomalies imply application; thus when you specify these in the attack object, application identification is applied.
- If you configure a specific application in a policy, you overwrite the application binding specified in an attack object. Table 63 on page 551 summarizes the binding with the application configuration in the IDP policy.

Table 63: Application Configuration in an IDP Policy

Application Type in the Policy	Binding Behavior	Application Identification
Default	Binds to the application or service configured in the attack object definition.	<ul style="list-style-type: none"> • Enabled for application-based attack objects • Disabled for service-based attack objects
Specific application	Binds to the application specified in the attack object definition.	Disabled
Any	Binds to all applications.	Disabled

- If you specify an application in an IDP policy, the application type configured in the attack object definition and in the IDP policy must match. The policy rule cannot specify two different applications (one in the attack object and the other in the policy).

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP Policies Overview on page 463
- Understanding the IDP Signature Database on page 535
- Understanding IDP Application Identification on page 549
- Disabling Junos OS Application Identification (CLI Procedure) on page 775
- Example: Configuring IDP Policies for Application Identification (CLI) on page 551

Example: Configuring IDP Policies for Application Identification (CLI)

For application identification to work, you must choose the **default** configuration option as the application type in an intrusion detection and prevention (IDP) policy rule. If you specify an application instead, the application identification feature is disabled and IDP matches the traffic with the specified application.

Before you begin, make sure that you have completed following:

1. Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Download the application package. See “Updating Junos OS Application Identification Extracted Application Package Overview” on page 771.

To configure an IDP policy for application identification:

1. Create an IDP policy, associate a rulebase with the policy, and define rules in the rulebase. The following statement creates an IDP policy **ABC** and defines rule **123** in the IPS rulebase:

```
user@host# set security idp idp-policy ABC rulebase-ips rule 123
```

2. Specify the application type as a match condition in the policy. The following statement specifies **default** as the application type:

```
user@host# set security idp idp-policy ABC rulebase-ips rule 123 match application default
```

3. Continue to configure other match conditions and actions for the policy. See “Example: Defining Rules for an IDP IPS Rulebase” on page 481.
4. If you are finished configuring the device, commit the configuration.
5. From configuration mode in the CLI, enter the **show security idp** command to verify the configuration. For more information, see the *Junos OS CLI Reference*.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP Policies Overview on page 463
- Understanding Junos OS Application Identification Application Package on page 770
- Understanding IDP Application Identification on page 549
- Verifying Application System Cache Statistics on page 556
- Disabling Junos OS Application Identification (CLI Procedure) on page 775
- Example: Setting Memory and Session Limits for IDP Application Identification (CLI) on page 558

Disabling Application Identification for an IDP Policy (CLI Procedure)

Application identification is enabled by default. You can disable application identification with the CLI.

To disable and application identification:

1. Specify the **disable** configuration option.

```
user@host# set security idp sensor-configuration application-identification disable
```


2. If you want to reenable application identification, delete the configuration statement that specifies disabling of application identification.

user@host# delete security idp sensor-configuration application-identification disable

3. If you are finished configuring the device, commit the configuration.
4. From configuration mode in the CLI, enter the **show security idp** command to verify the configuration. For more information, see the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP Policies Overview on page 463
 - Understanding the IDP Signature Database on page 535
 - Understanding IDP Application Identification on page 549
 - Verifying Application System Cache Statistics on page 556
 - Example: Configuring IDP Policies for Application Identification (CLI) on page 551
 - Example: Setting Memory and Session Limits for IDP Application Identification (CLI) on page 558

IDP Application Identification for Nested Applications

- Understanding IDP Application Identification for Nested Applications on page 553
- Activating IDP Application Identification for Nested Applications (CLI Procedure) on page 554
- Example: Adding IDP Application Information to Attack Logging for Nested Applications (CLI) on page 554

Understanding IDP Application Identification for Nested Applications

With the greater use of application protocol encapsulation, the need arises to support the identification of multiple different applications running on the same Layer 7 protocols. For example, applications such as Facebook and Yahoo Messenger can both run over HTTP, but there is a need to identify them as two different applications running on the same Layer 7 protocol. In order to do this, the current application identification layer is split into two layers: Layer 7 applications and Layer 7 protocols.

Included predefined application signatures have been created to detect the Layer 7 applications whereas the existing Layer 7 protocol signatures still function in the same manner. These predefined application signatures can be used in attack objects.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Application Identification on page 549
 - Activating IDP Application Identification for Nested Applications (CLI Procedure) on page 554
 - Example: Adding IDP Application Information to Attack Logging for Nested Applications (CLI) on page 554

Activating IDP Application Identification for Nested Applications (CLI Procedure)

Application identification for nested applications is turned on by default. You can manually turn this identification off by using the CLI.

```
user@host# edit security idp sensor-configuration application-identification
no-nested-application-identification
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Application Identification on page 549
 - Activating IDP Application Identification for Nested Applications (CLI Procedure) on page 554
 - Understanding IDP Application Identification for Nested Applications on page 553

Example: Adding IDP Application Information to Attack Logging for Nested Applications (CLI)

Nested application information added to IDP attack logging after “service” and before “rule” provides information on detected Layer 7 applications. In the following example, “Facebook” appears in the log file as nested application information.

```
Aug 29 20:46:32 4.0.0.254 RT_IDP: IDP_ATTACK_LOG_EVENT: IDP: at 1251603992, SIG
Attack log <4.0.0.1:33000->5.0.0.1:210> for TCP protocol and service SERVICE_IDP
application FACEBOOK by rule 1 of rulebase IPS in policy idpengine. attack: repeat=0,
action=NONE, severity=MEDIUM, name=http-url-attack-test, NAT
<8.11.163.220:0->0.0.0.0:0>, time-elapsed=0, inbytes=0, outbytes=0, inpackets=0,
outpackets=0, intf:untrust:ge-0/0/1.0->trust:ge-0/0/0.0, and misc-message -
```



NOTE: For further information on IDP logging, refer to “Understanding IDP Logging” on page 571.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Application Identification on page 549
 - Understanding Junos OS Application Identification for Nested Applications on page 776
 - Activating Junos OS Application Identification for Nested Applications (CLI Procedure) on page 776

IDP Application System Cache

- Understanding the IDP Application System Cache on page 555
- Understanding IDP Application System Cache Information for Nested Application Identification on page 555
- Deactivating IDP Application System Cache Information for Nested Application Identification (CLI Procedure) on page 556
- Verifying Application System Cache Statistics on page 556

Understanding the IDP Application System Cache

Application system cache (ASC) saves the mapping between an application type and the corresponding destination IP address, destination port, protocol type, and service.

Once an application is identified, its information is saved in the cache so that only one pattern matching is required for an application running on a particular system, thereby expediting the identification process.

A mapping is saved in the cache only if the matched signature contains both client-to-server and server-to-client patterns. This process protects the system from hackers who might send malicious packets through a legitimate server port so that it is interpreted as a different application.

By default, the application system cache saves the mapping information for 3600 seconds. However, you can configure the cache timeout value by using the CLI.

To minimize the impact on performance, application system cache is refreshed only when Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) traffic triggers a cache lookup. Without a cache lookup, the entries in the application system cache remain unchanged even after cache timeout.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Application Identification on page 549
 - Understanding IDP Application Identification for Nested Applications on page 553
 - Understanding IDP Application System Cache Information for Nested Application Identification on page 555
 - Deactivating IDP Application System Cache Information for Nested Application Identification (CLI Procedure) on page 556

Understanding IDP Application System Cache Information for Nested Application Identification

Nested application identification information is saved in the application system cache to improve performance. This cache is updated when a different application is identified. The only circumstances in which nested application information is not cached are the following:

- The application system cache is turned off for nested application identification.
- The matched application signatures have only client-to-server members.
- There is no valid server-to-client response seen for a transaction. This is done to prevent an attacker from sending invalid client-to-server requests to poison the application system cache.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Application Identification on page 549
 - Understanding Junos OS Application Identification for Nested Applications on page 776

- Understanding the Application System Cache on page 783
- Deactivating IDP Application System Cache Information for Nested Application Identification (CLI Procedure) on page 556

Deactivating IDP Application System Cache Information for Nested Application Identification (CLI Procedure)

Caching for nested applications is turned on by default. You can manually turn this caching off using the CLI.

```
user@host# edit security idp sensor-configuration application-identification
no-nested-application-system-cache
```

When you use the show command for the application system cache, nested application information is displayed as follows:

```
user@host# show security idp application-identification application-system-cache

Vsys-ID IP address Port Protocol Service Application
0 5.0.0.1 80 TCP HTTP FACEBOOK
0 5.0.0.2 80 TCP HTTP NONE
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Application Identification on page 549
 - Understanding Junos OS Application Identification for Nested Applications on page 776
 - Understanding the Application System Cache on page 783
 - Deactivating IDP Application System Cache Information for Nested Application Identification (CLI Procedure) on page 556

Verifying Application System Cache Statistics

Purpose Verify the application system cache (ASC) statistics.



NOTE: The application system cache will display the cache for application identification applications and nested applications.

Action From CLI operation mode, enter the **show services application-identification application-system-cache** command.

Sample Output

```
user@host> show services application-identification application-system-cache
Application System Cache Statistics:
Vsys-ID IP Address  Port  Protocol  Service  Application
0      5.0.0.1    65532 TCP      FTP      Unknown
Application System Cache statistics:

  Vsys-ID    IP address    Port  Protocol  Service  Application
0      5.0.0.1    65532 tcp      FTP      Unknown
0      20.0.0.4    23      tcp      TELNET   Unknown
0      20.0.0.6    23      tcp      TELNET   Unknown
```

0	20.0.0.2	23	tcp	TELNET	Unknown
0	20.0.0.2	25	tcp	SMTP	Unknown
0	20.0.0.6	25	tcp	SMTP	Unknown
0	20.0.0.4	25	tcp	SMTP	Unknown
0	20.0.0.3	135	tcp	MSRPC	Unknown
0	20.0.0.5	139	tcp	SMB	Unknown
0	20.0.0.7	139	tcp	SMB	Unknown
0	20.0.0.3	143	tcp	IMAP	Unknown
0	20.0.0.5	143	tcp	IMAP	Unknown
0	20.0.0.3	139	tcp	SMB	Unknown
0	20.0.0.7	143	tcp	IMAP	Unknown
0	20.0.0.3	80	tcp	HTTP	Unknown
0	20.0.0.5	80	tcp	HTTP	FACEBOOK
0	20.0.0.7	80	tcp	HTTP	ORKUT

Meaning The output shows a summary of the ASC statistics information. Verify the following information:

- Vsys-ID—Displays the virtual system identification number.
- IP address—Displays the destination address.
- Port—Displays the destination port on the server.
- Service—Displays the name of the service or application identified on the destination port.

For a complete description of **show security idp application-identification application-system-cache** output, see the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding the Application System Cache on page 783
 - Disabling Junos OS Application Identification (CLI Procedure) on page 775

IDP Memory and Session Limits

- Understanding Memory and Session Limit Settings for IDP Application Identification on page 557
- Example: Setting Memory and Session Limits for IDP Application Identification (CLI) on page 558

Understanding Memory and Session Limit Settings for IDP Application Identification

Although you cannot create application signatures with the IDP signature database, you can configure sensor settings to limit the number of sessions running application identification and also limit memory usage for application identification.

- Memory limit for a session—You can configure the maximum amount of memory bytes that can be used to save packets for application identification for one TCP or UDP session. You can also configure a limit for global memory usage for application

identification. Application identification is disabled for a session after the system reaches the specified memory limit for the session. However, IDP continues to match patterns. The matched application is saved to cache so that the next session can use it. This protects the system from attackers trying to bypass application identification by purposefully sending large client-to-server packets.

- **Number of sessions**—You can configure the maximum number of sessions that can run application identification at the same time. Application identification is disabled after the system reaches the specified number of sessions. You limit the number of sessions so that you can prevent a denial-of-service (DOS) attack, which occurs when too many connection requests overwhelm and exhaust all the allocated resources on the system.

Table 64 on page 558 provides the capacity of a central point (CP) session numbers for SRX3400, SRX3600, SRX5600, and SRX5800 devices.

Table 64: Maximum CP Session Numbers

SRX Series Devices	Maximum Sessions	Central Point (CP)
SRX3400	2.25 million	Combo-mode CP
SRX3600	2.25 million	Combo-mode CP
SRX5600	9 million	Full CP
	2.25 million	Combo-mode CP
SRX5800	10 million	Full CP
	2.25 million	Combo-mode CP

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Junos OS Application Identification Services on page 769
 - IDP Policies Overview on page 463
 - Understanding the IDP Signature Database on page 535
 - Example: Updating the IDP Signature Database Manually (CLI) on page 543
 - Example: Setting Memory and Session Limits for IDP Application Identification (CLI) on page 558

Example: Setting Memory and Session Limits for IDP Application Identification (CLI)

The configuration instructions in this topic describe how to configure memory and session limits for application identification.

Before you begin, make sure that you have completed following:

1. Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

2. Download the signature database. See “Example: Updating the IDP Signature Database Manually (CLI)” on page 543.

In the configuration instructions for this example, you configure the limit so that only 600 sessions can run application identification at the same time. You also configure 5000 memory bytes as the maximum amount of memory that can be used for saving packets for application identification for one TCP session.

To configure memory and session limits for application identification:

1. Specify the session limit for application identification. In the following statement you set the maximum number of sessions that can run application identification at the same time as **600**:

```
user@host# set security idp sensor-configuration application-identification
max-sessions 600
```

2. Specify the memory limit for application identification. In the following statement you configure a maximum of **5000** memory bytes to save packets for application identification:

```
user@host# set security idp sensor-configuration application-identification
max-tcp-session-packet-memory 5000
```

3. If you are finished configuring the device, commit the configuration.
4. From configuration mode in the CLI, enter the **show security idp** command to verify the configuration. For more information, see the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Application Identification on page 549
 - IDP Policies Overview on page 463
 - Understanding the IDP Signature Database on page 535
 - Example: Updating the IDP Signature Database Manually (CLI) on page 543
 - Understanding Memory and Session Limit Settings for IDP Application Identification on page 557

Verifying IDP Counters for Application Identification Processes

Purpose	Verify the IDP counters for the application identification processes.
Action	From the CLI, enter the show security idp counters application-identification command.
Sample Output	<pre>user@host> show security idp counters application-identification IDP counters: IDP counter type Value AI cache hits 2682 AI cache misses 3804 AI matches 74 AI no-matches 27 AI-enabled sessions 3804</pre>

AI-disabled sessions	2834
AI-disabled sessions due to cache hit	2682
AI-disabled sessions due to configuration	0
AI-disabled sessions due to protocol remapping	0
AI-disabled sessions due to non-TCP/UDP flows	118
AI-disabled sessions due to no AI signatures	0
AI-disabled sessions due to session limit	0
AI-disabled sessions due to session packet memory limit	34
AI-disabled sessions due to global packet memory limit	0

Meaning The output shows a summary of the application identification counters. Verify the following information:

- AI cache hits—Displays the number of hits on the application identification cache
- AI cache misses—Displays the number of times the application matches but the application identification cache entry is not added.
- AI matches—Displays the number of times the application matches, and an application identification cache entry is added.
- AI no-matches—Displays the number of times when application does not match.
- AI-enabled sessions—Displays the number of sessions on which application identification is enabled.
- AI-disabled sessions—Displays the number of sessions on which application identification is enabled.
- AI-disabled sessions due to cache hit—Displays the number of sessions on which application identification is disabled after a cache entry is matched. Application identification process is discontinued for this session.
- AI-disabled sessions due to configuration—Displays the number of sessions on which application identification is disabled because of the sensor configuration.
- AI-disabled sessions due to protocol remapping—Displays the number of sessions for which application identification is disabled because you have configured a specific service in the IDP policy rule definition.
- AI-disabled sessions due to non-TCP/UDP flows—Displays the number of sessions for which application identification is disabled because the session is not a TCP or UDP session.
- AI-disabled sessions due to no AI signatures—Displays the number of sessions for which application identification is disabled because no match is found on the application identification signatures.
- AI-disabled due to session limit—Displays the number of sessions for which application identification is disabled because sessions have reached the maximum limit configured. Application identification is disabled for future sessions too.
- AI-disabled due to session packet memory limit—Displays the sessions for which application identification is disabled because sessions have reached the maximum

memory limit on TCP or UDP flows. Application identification is disabled for future sessions too.

- AI-disabled due to global packet memory limit—Displays the sessions for which application identification is disabled because the maximum memory limit is reached. Application identification is disabled for future sessions too.

For a complete description of **show security idp counters** output, see the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Application Identification on page 549
 - Example: Setting Memory and Session Limits for IDP Application Identification (CLI) on page 558
 - Understanding IDP Service and Application Bindings by Attack Objects on page 550
 - Verifying Application System Cache Statistics on page 556

CHAPTER 26

IDP SSL Inspection

- IDP SSL Overview on page 563
- Supported IDP SSL Ciphers on page 564
- Understanding IDP Internet Key Exchange on page 565
- Understanding IDP SSL Server Key Management and Policy Configuration on page 566
- Displaying IDP SSL Keys and Associated Servers on page 566
- Adding IDP SSL Keys and Associated Servers on page 567
- Deleting IDP SSL Keys and Associated Servers on page 567
- Configuring an IDP SSL Inspection (CLI Procedure) on page 568

IDP SSL Overview

Secure Sockets Layer (SSL), also called Transport Layer Security (TLS), is a protocol suite for Web security that provides authentication, confidentiality and message integrity. Authentication guards against fraudulent transmissions by enabling a Web browser to validate the identity of a webserver. Confidentiality mechanisms ensure that communications are private. SSL enforces confidentiality by encrypting data to prevent unauthorized users from eavesdropping on electronic communications. Finally, message integrity ensures that the contents of a communication have not been tampered with.

Each SSL session begins with a handshake during which the client and server agree on the specific security key and the encryption algorithms to use for that session. At this time, the client also authenticates the server. Optionally, the server can authenticate the client. Once the handshake is complete, transfer of encrypted data can begin.

Juniper Networks provides Intrusion Detection and Prevention (IDP) SSL inspection that uses the SSL protocol suite consisting of different SSL versions, ciphers, and key exchange methods. Combined with the Application Identification feature, the SSL Inspection feature enables SRX Series devices to inspect HTTP traffic encrypted in SSL on any port. The following SSL protocols are supported:

- SSLv2
- SSLv3
- TLS

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP Policies Overview on page 463
 - Supported IDP SSL Ciphers on page 564
 - Understanding IDP Internet Key Exchange on page 565
 - Understanding IDP SSL Server Key Management and Policy Configuration on page 566
 - Configuring an IDP SSL Inspection (CLI Procedure) on page 568

Supported IDP SSL Ciphers

An SSL cipher comprises encryption cipher, authentication method, and compression. Junos OS supports all OPENSSL supported ciphers that do not involve the use of temporary private keys. For authentication, NULL, MD5, and SHA-1 authentication methods are supported.



NOTE: Compression and SSLv2 ciphers are not supported. Currently, most SSL servers automatically upgrade to a TLS cipher when an SSLv2 cipher is received in a client “hello” message. Check your browser to see how strong the ciphers can be and which ones your browser supports. (If the cipher is not in the list of supported ciphers, the session is ignored for deep packet inspection.)

Table 65 on page 564 shows the encryption algorithms supported by the SRX Series devices.

Table 65: Supported Encryption Algorithms

Cipher	Exportable	Type	Key Material	Expanded Key Material	Effective Key Bits	IV Size
NULL	No	Stream	0	0	0	N/A
DES-CBC-SHA	No	Block	8	8	56	8
DES-CBC3-SHA	No	Block	24	24	168	8
AES128-SHA	No	Block	16	16	128	16
AES256-SHA	No	Block	32	32	256	16

For more information on encryption algorithms, see “VPN Overview” on page 355. Table 66 on page 564 shows the supported SSL ciphers.

Table 66: Supported SSL Ciphers

Cipher Suites	Value
---------------	-------

Table 66: Supported SSL Ciphers (*continued*)

TLS_RSA_WITH_NULL_MD5	0x0001
TLS_RSA_WITH_NULL_SHA	0x0002
TLS_RSA_WITH_DES_CBC_SHA	0x0009
TLS_RSA_WITH_3DES_EDE_CBC_SHA	0x000A
TLS_RSA_WITH_AES_128_CBC_SHA	0x002F
TLS_RSA_WITH_AES_256_CBC_SHA	0x0035



NOTE: RC4 and IDEA ciphers are not supported because of license and OPENSSL library availability.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP SSL Overview on page 563
- Understanding IDP Internet Key Exchange on page 565
- Understanding IDP SSL Server Key Management and Policy Configuration on page 566

Understanding IDP Internet Key Exchange

Internet Key Exchange establishes a premaster secret that is used to generate symmetric keys for bulk data encryption and authentication. Section F.1.1 of RFC 2246 defines Transport Layer Security (TLS) authentication and key exchange methods. The two key exchange methods are:

- **RSA**—A key exchange algorithm that governs the way participants create symmetric keys or a secret that is used during an SSL session. RSA key exchange algorithm is the most commonly used method.
- **Diffie-Hellman**—A Diffie-Hellman (DH) key exchange method allows the participants to produce a shared secret value. The strength of the technique is that it allows the participants to create the secret value over an unsecured medium without passing the secret value through the wire.

Both RSA and Diffie-Hellman key exchange methods can use either a fixed or a temporary server key. IDP can successfully retrieve the premaster secret only if a fixed server key is used. Junos OS supports only the RSA key exchange method. For more information on Internet Key Exchange, see “Understanding Certificates” on page 386.



NOTE: Juniper IDP does not decrypt SSL sessions that use Diffie-Hellman key exchange.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP SSL Overview on page 563

- Supported IDP SSL Ciphers on page 564
- Understanding IDP SSL Server Key Management and Policy Configuration on page 566
- Configuring an IDP SSL Inspection (CLI Procedure) on page 568

Understanding IDP SSL Server Key Management and Policy Configuration

The device can support up to 1000 server private keys. Each key can have up to 100 servers that use it. This capacity is the same regardless of the number of SPUs available on the device because essentially each SPU needs to be able to access all the keys.

Multiple servers can share the same private key; however, one server can have only one private key. SSL decryption is disabled by default. Both plain and encrypted keys are supported.



NOTE: Junos OS does not encrypt SSL keys file.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP SSL Overview on page 563
- Displaying IDP SSL Keys and Associated Servers on page 566
- Adding IDP SSL Keys and Associated Servers on page 567
- Deleting IDP SSL Keys and Associated Servers on page 567
- Configuring an IDP SSL Inspection (CLI Procedure) on page 568

Displaying IDP SSL Keys and Associated Servers

- To display all installed server keys and associated server, use the following CLI command:

```
user@host> show security idp ssl-inspection key
```

Displays all server keys and IP addresses bound to those keys. The following example shows CLI output when the **show security idp ssl-inspection key** command is used:

```
Total SSL keys : 2
SSL server key and ip address :
Key : key1, server : 1.1.1.1
Key : key2, server : 2.2.2.2
Key : key2, server : 2.2.2.3
```

- To display IP addresses bound to a specific key, use the following CLI command:

```
user@host> show security idp ssl-inspection key <key-name>
```

The following is an example of the CLI output received when the **show security idp ssl-inspection key <key-name>** command is used:

```
Key : key1, server : 1.1.1.1
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP SSL Server Key Management and Policy Configuration on page 566
 - Adding IDP SSL Keys and Associated Servers on page 567
 - Deleting IDP SSL Keys and Associated Servers on page 567

Adding IDP SSL Keys and Associated Servers

When you are installing a key, you can password protect the key and also associate it to a server.

To install a Privacy-Enhanced Mail (PEM) key, use the following CLI command:

```
user@host> request security idp ssl-inspection key add <key-name> [file <file-path>]
server <server-ip> [password <password-string>]
```



NOTE: In a two-node SRX cluster, the key has to be manually copied over to both Node 0 and Node 1 at the same location for the request command to be successful.

You can also associate the key with a server at a later time by using the **add server** CLI command. A server can be associated with only one key. To associate a server to the installed key, use the following CLI command:

```
user@host> request security idp ssl-inspection key add <key-name> server <server-ip>
```



NOTE: The maximum key name length is 32 bytes, including the ending “\0”.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP SSL Server Key Management and Policy Configuration on page 566
 - Displaying IDP SSL Keys and Associated Servers on page 566
 - Deleting IDP SSL Keys and Associated Servers on page 567

Deleting IDP SSL Keys and Associated Servers

- To delete all keys and servers, use the following CLI command:

```
user@host> request security idp ssl-inspection key delete
```

All installed keys are deleted along with any associated servers.

- To delete a specific key and all associated servers with that key, use the following CLI command:

```
user@host> request security idp ssl-inspection key delete <key-name>
```

Deletes the specified key and all servers associated with that key.

- To delete a single server, use the following CLI command:

```
user@host> request security idp ssl-inspection key delete <key-name> server  
<server-ip>
```

Deletes the specified server that is bound to the specified key.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP SSL Server Key Management and Policy Configuration on page 566
 - Displaying IDP SSL Keys and Associated Servers on page 566
 - Adding IDP SSL Keys and Associated Servers on page 567

Configuring an IDP SSL Inspection (CLI Procedure)

SSL decoder is enabled by default. If you need to manually enable it via CLI, use the following CLI command.

```
set security idp sensor-configuration detector protocol-name SSL tunable-name sc_ssl_flags  
tunable-value 1
```

To configure an IDP SSL inspection, use the following CLI procedure:

```
[edit security]  
idp {  
  sensor-configuration {  
    ssl-inspection {  
      sessions <number>;  
    }  
  }  
}
```

The sensor now inspects traffic for which it has a key/server pair.



NOTE: Maximum supported sessions per SPU: default value is 10,000 and range is 1 to 100,000. The session limit is per SPU, and it is the same regardless of the number of SPUs on the device.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP SSL Overview on page 563
 - Understanding IDP Internet Key Exchange on page 565
 - Understanding IDP SSL Server Key Management and Policy Configuration on page 566

IDP Performance and Capacity Tuning

- Performance and Capacity Tuning for IDP Overview on page 569
- Configuring Session Capacity for IDP (CLI Procedure) on page 570

Performance and Capacity Tuning for IDP Overview

This topic provides an overview on performance and capacity tuning for an Intrusion Detection and Prevention (IDP) session.

If you are deploying IDP policies, you can configure the device to increase IDP session capacity. By using the provided commands to change the way the system allocates resources, you can achieve higher IDP session capacity.

By using the **maximize-idp-sessions** command, you can increase the IDP session capacity. In this mode, by default, the device assigns a greater weight value to firewall functions. Based on your IDP policy, you can shift the weight to IDP functions to maximize IDP performance. By shifting weight, you are increasing capacity and allocating more processing power for the given service.



NOTE: You should not configure the device to increase IDP session capacity if you are not using an IDP policy.

The device ships with an implicit default session capacity setting. This default value adds weight to firewall sessions. You can manually override the default by adding the **maximize-idp-sessions** setting to your configuration. When you do this, in addition to IDP session scaling, you can choose to assign weight values of equal, firewall, or IDP to firewall and IDP functions. Typically, when you only include IDP-recommended attacks or client-to-server attacks in your IDP policy, IDP functions consume less CPU resources, for this reason, you would select weight firewall to maximize device performance. Alternatively, if you add server-to-client attacks to your IDP policy, IDP functions consume higher CPU resources. For this reason, you would select weight IDP to maximize performance. Essentially, you will need to configure the weight based on the desired IDP policy and performance. You do this by examining the CPU resource utilization on the packet forwarding engine by using the command **show security monitoring fpc number**. For details on this command, see the *Junos OS CLI Reference*.

Related Topics • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- IDP Policies Overview on page 463
- Configuring Session Capacity for IDP (CLI Procedure) on page 570

Configuring Session Capacity for IDP (CLI Procedure)

The configuration instructions in this topic describe how modify session capacity for IDP policies.

You do this by adding the **maximize-idp-sessions** command and then adding the weight option to specify IDP sessions.



NOTE: The weight option depends on the **maximize-idp-sessions** command being set.

1. If you have an active IDP policy, you can configure the device to increase IDP session capacity by entering following command:

```
user@host# set security forwarding-process application-services  
maximize-idp-sessions
```

2. You can further adjust the weight of the firewall and IDP processing functions, such as in the case of heavier IDP policies with the following command:.

```
user@host# set security forwarding-process application-services  
maximize-idp-sessions weight idp
```

3. Commit your changes. You must reboot the device for any session capacity setting changes to take effect.



NOTE: If the device has **maximize-idp-sessions weight** enabled for IDP, and you do not have an IDP policy configured, a warning message appears when you commit your configuration. If you see this warning, you should remove your configured settings.

To turn **maximize-idp-sessions** settings off, remove the **maximize-idp-sessions** configuration.



NOTE: You must reboot the device for any **maximize-idp-sessions** setting changes to take effect.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- IDP Policies Overview on page 463
- Performance and Capacity Tuning for IDP Overview on page 569

CHAPTER 28

IDP Logging

- Understanding IDP Logging on page 571
- Understanding Application-Level DDoS Logging on page 572
- Enabling Attack and IP-Action Logging (CLI Procedure) on page 573
- IDP Log Suppression Attributes on page 574
- Understanding IDP Log Information Usage on the Infranet Controller on page 576
- Security Packet Capture on page 577

Understanding IDP Logging

The basic Junos OS system logging continues to function after Intrusion Detection and Prevention (IDP) is enabled. An IDP-enabled device continues to record events that occur because of routine operations, such as a user login into the configuration database. It records failure and error conditions, such as failure to access a configuration file. You can configure files to log system messages and also assign attributes, such as severity levels, to messages. In addition to the regular system log messages, IDP generates event logs for attacks. For information about monitoring events, managing system log files, and configuring packet capture see the *Junos OS Administration Guide for Security Devices*.

IDP generates event logs when an event matches an IDP policy rule in which logging is enabled. When you configure a rule for logging, the device creates a log entry for each event that matches that rule. You can use the CLI or J-Web to configure the policy rules to generate event logs.

Because IDP event logs are generated during an attack, log generation happens in bursts, generating a much larger volume of messages during an attack. In comparison to other event messages, the message size is also much larger for attack generated messages. The log volume and message size are important concerns for log management. To better manage the volume of log messages, IDP supports log suppression.

By configuring log suppression you can suppress multiple instances of the same log occurring from the same or similar sessions over the same period of time. Enabling log suppression ensures that minimal numbers of logs are generated for the same event or attack that occurs multiple times.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP Policies Overview on page 463

- Understanding IDP Policy Rules on page 470
- Understanding IDP Log Suppression Attributes on page 574
- Understanding IDP Log Information Usage on the Infranet Controller on page 576
- Example: Defining Rules for an IDP IPS Rulebase on page 481

Understanding Application-Level DDoS Logging

Intrusion Detection and Prevention (IDP) generates three types of application-level distributed denial-of-service (application-level DDoS) event logs: attack, state transition, and ip-action. These event logs provide visibility into the application-level DDoS state and provide notifications on occurrences of application-level DDoS attacks for each protected application server.

IDP generates application-level DDoS attack event logs when logging is enabled and an event matches an application-level DDoS policy rule. When you configure a rule with logging enabled, the device creates a log entry for each attack event that matches the rule. For more information about the application-level DDoS rulebase, see “Understanding IDP Application-Level DDoS Rulebases” on page 479.

The attack event log contains the following information:

- Time generated (the date/time in which the log is generated)
- Ingress and egress zone and interface information
- Sources and destination IP address and port numbers
- Connection, context, and context value rates
- Time-binding information
- Policy name
- Rulebase name and rule name
- application-level DDoS application name
- Layer 4 protocol
- Application service (such as DNS and HTTP)
- Context and value rates
- Context value (presented in ASCII and hexadecimal formats)
- Action taken on the event

To reduce the volume of application-level DDoS attack event logs, when you configure an application-level DDoS application with time-binding-count in a rule that has logging enabled, IDP generates an application-level DDoS attack event log only when an attack is detected for time-binding-count times for each time-binding-period seconds. Without time-binding-count configured for an application-level DDoS application, IDP generates an application-level DDoS attack event log for each detected attack, and these logs are

subjected to log suppression. The repeat-count field in the log represents how many times this log event would have been sent if log suppression was applied.

IDP generates application-level DDoS state transition event logs when the number of application transactions exceeds or falls behind the configured connection or context hit rate thresholds. State transition event logs are enabled by default, and IDP generates state transition event logs based on user-configured connection, context, or context value thresholds. IDP exhibits hysteresis for state transitions, due to this fact, the state transition log event is generated after incoming traffic connection or context rates have fallen behind by 20 percent (by default) of the configured threshold.



NOTE: State transition logging is enabled by default and cannot be enabled or disabled, it is part of the standard system logging.

The state event log contains the following information:

- Time generated (the date/time in which the log is generated)
- IP address of the protected server
- Port
- Interface and zone
- Policy name
- Rulebase name and rule name
- application-level DDoS application name
- Layer 4 protocol
- Application service (such as DNS and HTTP)
- Description of the transition event
- Description of the context value (presented in ASCII and hexadecimal formats)

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding IDP Log Suppression Attributes on page 574
- Understanding IDP Logging on page 571
- Understanding IDP Log Information Usage on the Infranet Controller on page 576
- IDP Application-Level DDoS Attack Overview on page 523

Enabling Attack and IP-Action Logging (CLI Procedure)

To enable attack and ip-action logging, perform the following steps:

Enable attack logs

```
[edit security idp]
```

```
user@host# set idp idp-policy AppDDoS-policy-name rulebase-ddos rule
AppDDoS-rule-name then notification log-attacks
```

Enable ip-action logs

```
[edit security idp]
user@host# set idp-policy AppDDoS-policy-name rulebase-ddos rule AppDDoS-rule-name
then ip-action log
```

Once enabled, the application-level DDoS logs will appear in the regular system logs. For information about monitoring events and managing system log files, see the *Junos OS Administration Guide for Security Devices*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Log Suppression Attributes on page 574
 - Understanding IDP Logging on page 571
 - Understanding IDP Log Information Usage on the Infranet Controller on page 576
 - IDP Application-Level DDoS Attack Overview on page 523

IDP Log Suppression Attributes

- Understanding IDP Log Suppression Attributes on page 574
- Example: Configuring IDP Log Suppression Attributes on page 575

Understanding IDP Log Suppression Attributes

Log suppression ensures that minimal numbers of logs are generated for the same event or attack that occurs multiple times. Log suppression is enabled by default. You can configure certain log suppression attributes to suppress logs according to your needs. When configuring log suppression, keep in mind that log suppression can negatively impact sensor performance if you set the reporting interval too high.

You can configure the following log suppression attributes:

- Include destination addresses while performing log suppression—You can choose to combine log records for events with a matching source address. By default, the IDP sensor does not consider destination when matching events for log suppression.
- Number of log occurrences after which log suppression begins—You can specify the number of instances that a specific event must occur before log suppression begins. By default, log suppression begins after the first occurrence.
- Maximum number of logs that log suppression can operate on—When log suppression is enabled, Intrusion Detection and Prevention (IDP) must cache log records so that it can identify when multiple occurrences of the same event occur. You can specify how many log records are tracked simultaneously by IDP. By default, the maximum number of log records that IDP can operate on is 16,384.
- Time after which suppressed logs are reported—When log suppression is enabled, IDP maintains a count of occurrences of the same event. After the specified number of

seconds have passed, IDP writes a single log entry containing the count of occurrences. By default, IDP reports suppressed logs after 5 seconds.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - IDP Policies Overview on page 463
 - Understanding IDP Policy Rules on page 470
 - Example: Configuring IDP Log Suppression Attributes on page 575

Example: Configuring IDP Log Suppression Attributes

This example shows how to configure log suppression attributes.

Requirements

Before you begin:

- Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
- Download the signature database. See “Updating the IDP Signature Database Manually Overview” on page 543.

Overview

Log suppression ensures that minimal numbers of logs are generated for the same event or attack that occurs multiple times. Log suppression is enabled by default. You can configure certain log suppression attributes to suppress logs according to your needs.

In this example, you configure log suppression to begin after the second occurrence of an event and specify that logs are reported after 20 seconds.

Configuration

Step-by-Step Procedure

To configure log suppression attributes:

1. Specify the log number after which you want to start log suppression.

```
[edit]
user@host# set security idp sensor-configuration log suppression start-log 2
```
2. Specify the maximum time after which suppressed logs are reported.

```
[edit]
user@host# set security idp sensor-configuration log suppression max-time-report 20
```
3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show security idp** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Updating the IDP Signature Database Manually Overview on page 543
 - Example: Defining Rules for an IDP IPS Rulebase on page 481
 - Understanding IDP Log Suppression Attributes on page 574

Understanding IDP Log Information Usage on the Infranet Controller

The infranet controller for the Unified Access Control (UAC) appliance can use Intrusion Detection and Prevention (IDP) attack log information sent from the Juniper Networks device to apply access policies for traffic in which IDP logs indicate an attack has been detected. Using a secure channel of communication, these IDP logs are sent to the infranet controller directly and securely. IDP attack logs are sent to the infranet controller through the JUEP communication channel.

This topic contains the following sections:

- Message Filtering to the Infranet Controller on page 576
- Configuring Infranet Controller Logging on page 576

Message Filtering to the Infranet Controller

When you configure the infranet controller to receive IDP log messages, you set certain filtering parameters on the infranet controller. Without this filtering, the infranet controller could potentially receive too many log messages. The filtering parameters could include the following:

- The infranet controller should only receive communications from IDP for sessions it has authenticated. See the *Unified Access Control Administration Guide* for details.
- You can create infranet controller filters for receiving IDP logs files based on the their severity. For example, if on the infranet controller the severity is set to high, then IDP only sends logs which have a severity greater than or equal to high. See the *Unified Access Control Administration Guide* for details.
- From the infranet controller, you can disable the receiving of all IDP logs. See the *Unified Access Control Administration Guide* for details.

Configuring Infranet Controller Logging

All the configuration for receiving and filtering IDP logs is done on the infranet controller. You should refer to the *Unified Access Control Administration Guide* for configuration information for receiving IDP logs and details on the JUEP communication channel.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding IDP Log Suppression Attributes on page 574

- Understanding IDP Logging on page 571
- Understanding Application-Level DDoS Logging on page 572

Security Packet Capture

- Understanding Security Packet Capture on page 577
- Example: Configuring Security Packet Capture (CLI) on page 577
- Example: Verifying Security Packet Capture (CLI) on page 578

Understanding Security Packet Capture

Viewing packets that precede and follow an attack helps you determine the purpose and extent of an attempted attack, whether an attack was successful, and if any network damage was caused by an attack. Packet analysis also aids in defining attack signatures to minimize false positives.

If packet capture is enabled when an attack is logged, a specified number of packets before and after the attack can be captured for the session. When all packets have been collected, they are transmitted in Device Management Interface (DMI) to a host device for offline analysis.

A notification option in the IDP policy rule enables packet capture when a rule match occurs. The option further defines the number of packets to be captured and the duration of packet capture for the associated session.

An IDP sensor configuration defines the device specifications for the packet capture. Options for this command determine the memory to be allocated for packet capture, and the source and host devices between which the packet capture object will be transmitted.

A **show** command displays packet capture counters that provide details about the progress, success, and failure of packet capture activity on the device.

Example: Configuring Security Packet Capture (CLI)

The following example configures a packet capture for rule 1 of policy pol0. The rule specifies that, if an attack occurs, 10 packets before the attack and 3 packets after the attack will be captured, and that the post-attack capture should time out after 60 seconds. The sensor configuration is modified to allocate 5% of available memory and 15% of the IDP sessions to packet capture. When the packet capture object is prepared, it is transmitted from device 10.56.97.3 to port 5 on device 10.24.45.7.

1. Navigate to the notification level for rule 1, policy pol0 in the configuration hierarchy.

```
[edit]
user@host# set security idp idp-policy pol0 rulebase-ips rule 1 then notification
```
2. Define the size and timing constraints for each packet capture:

```
[edit security idp idp-policy pol0 rulebase-ips rule 1 then notification]
user@host# set packet-log pre-attack 10 post-attack 3 post-attack-timeout 60
```

3. Navigate to the security idp sensor-configuration level of the configuration hierarchy:

```
[edit security idp idp-policy pol0 rulebase-ips rule 1 then notification]
user@host# top
[edit]
user@host# set security idp sensor-configuration
```
4. Allocate the device resources to be used for packet capture (5% of available device memory and 15% of the IDP sessions):

```
[edit security idp sensor-configuration]
user@host# set packet-log total-memory 5 max-sessions 15
```
5. Identify the source and host devices for transmitting the packet-capture object:

```
[edit security idp sensor-configuration]
user@host# set packet-log source-address 10.56.97.3 host 10.24.45.7 port 5
```
6. Navigate to the top of the hierarchy, and commit the configuration.

```
[edit security idp sensor-configuration]
user@host# top
[edit]
user@host# commit
```

For additional command options and default values, see the *Junos OS CLI Reference*.

For information about monitoring events and managing system log files, see the *Junos OS Administration Guide for Security Devices*.

Example: Verifying Security Packet Capture (CLI)

Monitor packet capture statistics issuing the following **show** command from the CLI prompt.

```
user@host> show security idp counters packet-log
```

IDP counters:	Value
Total packets captured since packet capture was activated	0
Total sessions enabled since packet capture was activated	0
Sessions currently enabled for packet capture	0
Packets currently captured for enabled sessions	0
Packet clone failures	0
Session log object failures	0
Session packet log object failures	0
Sessions skipped because session limit exceeded	0
Packets skipped because total memory limit exceeded	0

PART 8

Unified Threat Management

- Unified Threat Management Overview on page 581
- Antispam Filtering on page 587
- Full Antivirus Protection on page 605
- Express Antivirus Protection on page 649
- Content Filtering on page 665
- Web Filtering on page 679

CHAPTER 29

Unified Threat Management Overview

- Unified Threat Management Overview on page 581
- Understanding UTM Custom Objects on page 582
- UTM Licensing on page 582
- WELF Logging for UTM Features on page 583

Unified Threat Management Overview

Unified Threat Management (UTM) is a term used to describe the consolidation of several security features into one device, protecting against multiple threat types. The advantage of UTM is streamlined installation and management of these multiple security capabilities.

The security features provided as part of the UTM solution are:

- Antispam — E-mail spam consists of unwanted e-mail messages, usually sent by commercial, malicious, or fraudulent entities. The antispam feature examines transmitted e-mail messages to identify e-mail spam. When the device detects an e-mail message deemed to be spam, it either drops the message or tags the message header or subject field with a preprogrammed string. The antispam feature uses a constantly updated spam block list (SBL). Sophos updates and maintains the IP-based SBL. The antispam feature is a separately licensed subscription service.
- Full File-Based Antivirus — A virus is executable code that infects or attaches itself to other executable code to reproduce itself. Some malicious viruses erase files or lock up systems. Other viruses merely infect files and overwhelm the target host or network with bogus data. The full file-based antivirus feature provides file-based scanning on specific Application Layer traffic checking for viruses against a virus signature database. It collects the received data packets until it has reconstructed the original application content, such as an e-mail file attachment, and then scans this content. Kaspersky Lab provides the internal scan engine. The full file-based antivirus scanning feature is a separately licensed subscription service.
- Express Antivirus — Express antivirus scanning is offered as a less CPU intensive alternative to the full file-based antivirus feature. The express antivirus feature, like the full antivirus feature, scans specific Application Layer traffic for viruses against a virus signature database. However, unlike full antivirus, express antivirus does not reconstruct the original application content. Rather, it just sends (streams) the received data packets, as is, to the scan engine. With express antivirus, the virus scanning is

executed by a hardware pattern matching engine. This improves performance while scanning is occurring, but the level of security provided is lessened. Juniper Networks provides the scan engine. The express antivirus scanning feature is a separately licensed subscription service.

- **Content Filtering** — Content filtering blocks or permits certain types of traffic based on the MIME type, file extension, protocol command, and embedded object type. Content filtering does not require a separate license.
- **Web Filtering** — Web filtering lets you manage Internet usage by preventing access to inappropriate Web content. There are three types of Web filtering solutions. In the case of the integrated Web filtering solution, the decision-making for blocking or permitting Web access is done on the device after it identifies the category for a URL either from user-defined categories or from a category server (Websense provides the CPA Server). The integrated Web filtering feature is a separately licensed subscription service. The redirect Web filtering solution intercepts HTTP requests and forwards the server URL to an external URL filtering server provided by Websense to determine whether to block or permit the requested Web access. Redirect Web filtering does not require a separate license. With Juniper Local Web Filtering, the decision-making for blocking or permitting Web access is done on the device after it identifies the category for a URL from user-defined categories stored on the device. With Local filtering, there is no additional Juniper license or remote category server required.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Understanding UTM Custom Objects

Before you can configure most UTM features, you must first configure the custom objects for the feature in question. Custom objects are global parameters for UTM features. This means that configured custom objects can be applied to all UTM policies where applicable, rather than only to individual policies.

The following UTM features make use of certain custom objects:

- Anti-Virus (see “Full Antivirus Pattern Update Configuration Overview” on page 607)
- Web Filtering (see “Integrated Web Filtering Configuration Overview” on page 682)
- Anti-Spam (see “Server-Based Antispam Filtering Configuration Overview” on page 588)
- Content Filtering (see “Content Filtering Configuration Overview” on page 668)

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

UTM Licensing

- Understanding UTM Licensing on page 583
- Updating UTM Licenses (CLI Procedure) on page 583

Understanding UTM Licensing

The majority of UTM features function as a subscription service requiring a license. You can redeem this license once you have purchased your subscription license SKUs. You redeem your license by entering your authorization code and chassis serial number into the Customer Service LMS interface. Once your entitlement is generated, you can use the CLI from your device to send a license update request to the LMS server. The LMS server then sends your subscription license directly to the device.



NOTE: UTM requires 1 GB of memory. If your J2320, J2350, or J4350 device has only 512 MB of memory, you must upgrade the memory to 1 GB to run UTM.

Table 67: UTM Feature Subscription Service License Requirements

UTM Feature	Requires License
Antispam	Yes
Antivirus: full	Yes
Antivirus: express	Yes
Content Filtering	No
Web Filtering: integrated	Yes
Web Filtering: redirect	No
Web Filtering: local	No

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Updating UTM Licenses (CLI Procedure)

To apply your UTM subscription license to the device, use the following CLI command:

```
user@host> request system license update
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

WELF Logging for UTM Features

- Understanding WELF Logging for UTM Features on page 583
- Example: Configuring WELF Logging for UTM Features on page 584

Understanding WELF Logging for UTM Features

UTM features support the WELF standard. The WELF Reference defines the WebTrends industry standard log file exchange format. Any system logging to this format is compatible

with Firewall Suite 2.0 and later, Firewall Reporting Center 1.0 and later, and Security Reporting Center 2.0 and later.

A WELF log file is composed of records. Each record is a single line in the file. Records are always in chronological order. The earliest record is the first record in the file; the most recent record is the last record in the file. WELF places no restrictions on log filenames or log file rotation policies.



NOTE: Each WELF record is composed of fields. The record identifier field (**id=**) must be the first field in a record. All other fields can appear in any order.

The following is a sample WELF record:

```
id=firewall time="2000-2-4 12:01:01" fw=192.168.0.238 pri=6 rule=3 proto=http
src=192.168.0.23 dst=6.1.0.36 rg=www.webtrends.com/index.html op=GET result=0
rcvd=1426
```

The fields from the example WELF record include the following required elements (all other fields are optional):

- **id** (Record identifier)
- **time** (Date/time)
- **fw** (Firewall IP address or name)
- **pri** (Priority of the record)

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Configuring WELF Logging for UTM Features

This example shows how to configure WELF logging for UTM features.

- Requirements on page 584
- Overview on page 584
- Configuration on page 585
- Verification on page 586

Requirements

Before you begin, review the fields used to create a WELF log file and record. See “Understanding WELF Logging for UTM Features” on page 583.

Overview

A WELF log file is composed of records. Each record is a single line in the file. Records are always in chronological order. The earliest record is the first record in the file; the most recent record is the last record in the file. WELF places no restrictions on log filenames or log file rotation policies. In this example, the severity level is emergency and the name of the security log stream is **stream-utm-welf**.

Configuration

CLI Quick Configuration To quickly configure WELF logging for UTM features, copy the following commands and paste them into the CLI.

```
[edit]
set security log source-address 1.2.3.4 stream utm-welf
set security log source-address 1.2.3.4 stream utm-welf format welf
set security log source-address 1.2.3.4 stream utm-welf format welf category
content-security
set security log source-address 1.2.3.4 stream utm-welf format welf category
content-security severity emergency
set security log source-address 1.2.3.4 stream utm-welf format welf category
content-security severity emergency host 5.6.7.8
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure WELF logging for UTM features:

1. Set the security log source IP address.

```
[edit security log]
user@host# set source-address 1.2.3.4
```



NOTE: You must save the WELF logging messages to a dedicated WebTrends server.

2. Name the security log stream.

```
[edit security log]
user@host# set source-address 1.2.3.4 stream utm-welf
```

3. Set the format for the log messages.

```
[edit security log]
user@host# set source-address 1.2.3.4 stream utm-welf format welf
```

4. Set the category of log messages that are sent.

```
[edit security log]
user@host# set source-address 1.2.3.4 stream utm-welf format welf category
content-security
```

5. Set the severity level of log messages that are sent.

```
[edit security log]
user@host# set source-address 1.2.3.4 stream utm-welf format welf category
content-security severity emergency
```

6. Enter the host address of the dedicated WebTrends server to which the log messages are to be sent.

```
[edit security log]
```

```
user@host# set source-address 1.2.3.4 stream utm-welf format welf category
content-security severity emergency host 5.6.7.8
```

Results From configuration mode, confirm your configuration by entering the **show security log** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security log
stream utm-welf {
  severity emergency;
  format welf;
  category content-security;
  host {
    5.6.7.8;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Security Log on page 586

Verifying the Security Log

Purpose Verify that the WELF log for UTM features is complete.

Action From operational mode, enter the **show security utm status** command to verify if the UTM service is running or not.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Unified Threat Management Overview on page 581

CHAPTER 30

Antispam Filtering

- Antispam Filtering Overview on page 587
- Server-Based Spam Filtering on page 587
- Local List Spam Filtering on page 594
- Understanding Spam Message Handling on page 603

Antispam Filtering Overview

Spam consists of unwanted e-mail messages, usually sent by commercial, malicious, or fraudulent entities. The antispam feature examines transmitted e-mail messages to identify spam. When the device detects a message deemed to be spam, it blocks the e-mail message or tags the e-mail message header or subject with a preprogrammed string.

Antispam filtering allows you to use both a third-party server-based spam block list (SBL) and to optionally create your own local whitelists (benign) and blacklists (malicious) for filtering against e-mail messages. The antispam feature is not meant to replace your antispam server, but to complement it.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Server-Based Spam Filtering

- Understanding Server-Based Antispam Filtering on page 587
- Server-Based Antispam Filtering Configuration Overview on page 588
- Example: Configuring Server-Based Antispam Filtering on page 589

Understanding Server-Based Antispam Filtering



NOTE: Server-based spam filtering supports only IP-based spam block list blacklist lookup. Sophos updates and maintains the IP-based spam block list. Server-based antispam filtering is a separately licensed subscription service. When your antispam license key expires, you can continue to use locally defined black and whitelists.

Server-based antispam filtering requires Internet connectivity with the spam block list (SBL) server. Domain Name Service (DNS) is required to access the SBL server. The firewall performs SBL lookups through the DNS protocol. The lookups are against the IP address of the sender (or relaying agent) of the e-mail, adding the name of the SBL server as the authoritative domain. The DNS server then forwards each request to the SBL server, which returns a DNS response to the device. The device then interprets the DNS response to determine if the e-mail sender is a spammer.

IP addresses that are included in the block lists are generally considered to be invalid addresses for mail servers or easily compromised addresses. Criteria for listing an IP address as a spammer on the SBL can include:

- Running an SMTP open relay service
- Running open proxy servers (of various kinds)
- Being a zombie host possibly compromised by a virus, worm, Trojan, or spyware
- Using a dynamic IP range
- Being a confirmed spam source with a known IP address

By default, the device first checks incoming e-mail against local white and blacklists. If there are no local lists, or if the sender is not found on local lists, the device proceeds to query the SBL server over the Internet. When both server-based spam filtering and local list spam filtering are enabled, checks are done in the following order:

1. The local whitelist is checked. If there is a match, no further checking is done. If there is no match...
2. The local blacklist is checked. If there is a match, no further checking is done. If there is no match...
3. The SBL server list is checked.



NOTE: SBL server matching stops when the antispam license key is expired.

Related Topics

- [Server-Based Antispam Filtering Configuration Overview on page 588](#)
- [Example: Configuring Server-Based Antispam Filtering on page 589](#)
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

Server-Based Antispam Filtering Configuration Overview

For each UTM feature, configure feature parameters in the following order:

1. Configure UTM custom objects for the feature:

```
user@host# set security utm custom-objects
```
2. Configure the main feature parameters, using feature profiles.

```
user@host# set security utm feature-profile anti-spam
```

3. Configure a UTM policy for each protocol, and attach this policy to a profile.

```
user@host# set security utm utm-policy utmp1 anti-spam smtp-profile smtp1
```



NOTE: Antispam filtering is only supported for the SMTP protocol.

4. Attach the UTM policy to a security policy.

```
user@host# set security policies from-zone trust to-zone untrust policy p1 then permit
application-services utm-policy utmp1
```

Related Topics

- Understanding Server-Based Antispam Filtering on page 587
- Example: Configuring Server-Based Antispam Filtering on page 589
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Configuring Server-Based Antispam Filtering

This example shows how to configure server-based antispam filtering.

- Requirements on page 589
- Overview on page 589
- Configuration on page 589
- Verification on page 594

Requirements

Before you begin, review how to configure the feature parameters for each UTM feature. See “Server-Based Antispam Filtering Configuration Overview” on page 588.

Overview

Server-based antispam filtering requires Internet connectivity with the spam block list (SBL) server. Domain Name Service (DNS) is required to access the SBL server.

Configuration

CLI Quick Configuration

To quickly configure server-based antispam filtering, copy the following commands and paste them into the CLI.

```
[edit]
set security utm feature-profile anti-spam sbl profile sblprofile1 sbl-default-server
set security utm feature-profile anti-spam sbl profile sblprofile1 sbl-default-server
spam-action block
set security utm feature-profile anti-spam sbl profile sblprofile1 sbl-default-server
custom-tag-string ***spam***
set security utm utm-policy spampolicy1 anti-spam smtp-profile sblprofile1
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy1 match
source-address any
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy1 match
destination-address any
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy1 match
application junos-smtp
```

set security policies from-zone trust to-zone untrust policy utmsecuritypolicy1 then permit application-services utm-policy spampolicy1

J-Web Quick Configuration

To configure server-based antispam filtering:

1. Configure a profile and enable/disable the SBL server lookup. Select **Configure>Security>UTM>Anti-Spam**.
 - a. In the Anti-Spam profiles configuration window, click **Add** to configure a profile for the SBL server, or click **Edit** to modify an existing item.
 - b. In the Profile name box, enter a unique name for the antispam profile that you are creating.
 - c. If you are using the default server, select **Yes** next to Default SBL server. If you are not using the default server, select **No**.



NOTE: The SBL server is predefined on the device. The device comes preconfigured with the name and address of the SBL server. If you do not select Yes, you are disabling server-based spam filtering. You should disable it only if you are using only local lists or if you do not have a license for server-based spam filtering.

- d. In the Custom tag string box, enter a custom string for identifying a message as spam. By default, the devices uses ***SPAM***.
 - e. From the antispam action list, select the action that the device should take when it detects spam. Options include Tag subject, Block email, and Tag header.
2. Configure a UTM policy for SMTP, and attach the antispam profile to this UTM policy.
 - a. Select **Configure>Security>Policy>UTM Policies**.
 - b. In the UTM policy configuration window, click **Add**.
 - c. In the policy configuration window, select the **Main** tab.
 - d. In the Policy name box, type a unique name for the UTM policy.
 - e. In the Session per client limit box, type a session per client limit. Valid values range from 0 to 2000.
 - f. From the Session per client over limit list, select the action that the device should take when the session per client limit for this UTM policy is exceeded. Options include Log and permit and Block.
 - g. Select the **Anti-Spam profiles** tab in the pop-up window.
 - h. From the SMTP profile list, select an antispam profile to attach to this UTM policy.

3. Attach the UTM policy to a security policy.
 - a. Select **Configure>Security>Policy>FW Policies**.
 - b. In the Security Policy window, click **Add** to configure a security policy with UTM or click **Edit** to modify an existing policy.
 - c. In the Policy tab, type a name in the **Policy Name** box.
 - d. Next to From Zone, select a zone from the list.
 - e. Next to To Zone, select a zone from the list.
 - f. Choose a source address.
 - g. Choose a destination address.
 - h. Choose an application by selecting **junos-smtp** (for antispam) in the Application Sets box and move it to the Matched box.
 - i. Next to Policy Action, select one of the following: **Permit**, **Deny**, or **Reject**.



NOTE: When you select Permit for Policy Action, several additional fields become available in the Applications Services tab, including UTM Policy.

- j. Select the **Application Services** tab.
- k. Next to UTM Policy, select the appropriate policy from the list. This attaches your UTM policy to the security policy.
- l. Click **OK** to check your configuration and save it as a candidate configuration.
- m. If the policy is saved successfully, you receive a confirmation, and you must click **OK** again. If the profile is not saved successfully, click **Details** in the pop-up window to discover why.



NOTE:

- You must activate your new policy to apply it.
- In SRX devices the confirmation window that notifies you that the policy is saved successfully, disappears automatically.

- n. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure server-based antispam filtering:

1. Create a profile.

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile sblprofile1
```

2. Enable or disable the default SBL server lookup.

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile sblprofile1
sbl-default-server
```



NOTE: If you are using server-based antispam filtering, you should type `sbl-default-server` to enable the default SBL server. (The SBL server is predefined on the device. The device comes preconfigured with the name and address of the SBL server.) You should disable server-based antispam filtering using the `no-sbl-default-server` option only if you are using only local lists or if you do not have a license for server-based spam filtering.

3. Configure the action to be taken by the device when spam is detected (block, tag-header, or tag-subject).

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile sblprofile1 sbl-default-server
spam-action block
```

4. Configure a custom string for identifying a message as spam.

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile sblprofile1
sbl-default-server custom-tag-string ***spam***
```

5. Configure a UTM policy for SMTP to which you attach the antispam feature profile.

```
[edit security]
user@host# set utm utm-policy spampolicy1
```

6. Attach the spam feature profile to the UTM policy.

```
[edit security]
user@host# set utm utm-policy spampolicy1 anti-spam smtp-profile sblprofile1
```

7. Configure a security policy for UTM, and attach the UTM policy to the security policy.

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy1 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy1 match destination-address any
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy1 match application junos-smtp
```



```
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy1 then permit application-services utm-policy spampolicy1
```



NOTE: The device comes preconfigured with a default antispam policy. The policy is called `junos-as-defaults`. It contains the following configuration parameters:

```
anti-spam {
  sbl {
    profile junos-as-defaults {
      sbl-default-server;
      spam-action block;
      custom-tag-string "***SPAM***";
    }
  }
}
```

Results From configuration mode, confirm your configuration by entering the **show security utm** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security utm
feature-profile {
  anti-spam {
    sbl {
      profile sblprofile1 {
        sbl-default-server;
        spam-action block;
        custom-tag-string ***spam***;
      }
    }
  }
}
utm-policy spampolicy1 {
  anti-spam {
    smtp-profile sblprofile1;
  }
}
```

```
[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy utmsecuritypolicy1 {
    match {
      source-address any;
      destination-address any;
      application junos-smtp;
    }
    then {
      permit {
        application-services {
          utm-policy spampolicy1;
        }
      }
    }
  }
}
```

```
    }  
  }  
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Antispam Statistics on page 594

Verifying Antispam Statistics

Purpose Verify the antispam statistics.

Action From operational mode, enter the **show security utm anti-spam status** and **show security utm anti-spam statistics** commands.

The following information appears:

```
SBL Whitelist Server:  
SBL Blacklist Server:  
server.juniper.net  
DNS Server:  
Primary : 1.2.3.4, Src Interface: ge-0/0/0  
Secondary: 2.3.4.5, Src Interface: ge-0/0/1  
Ternary : 0.0.0.0, Src Interface: fe-0/0/2  
  
Total connections: #  
Denied connections: #  
Total greetings: #  
Denied greetings: #  
Total e-mail scanned: #  
Spam total: #  
Spam tagged: #  
Spam dropped: #  
DNS errors: #  
Timeout errors: #  
Return errors: #  
Invalid parameter errors: #  
Statistics start time:  
Statistics for the last 10 days.
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Server-Based Antispam Filtering on page 587

Local List Spam Filtering

- Understanding Local List Antispam Filtering on page 595
- Local List Antispam Filtering Configuration Overview on page 595
- Example: Configuring Local List Antispam Filtering on page 596

Understanding Local List Antispam Filtering

When creating your own local whitelist and blacklist for antispam filtering, you can filter against domain names, e-mail addresses, and/or IP addresses. Pattern matching works a bit differently depending upon the type of matching in question. For example, pattern matching for domain names uses a longest suffix match algorithm. If the sender e-mail address has a domain name of aaa.bbb.ccc, the device tries to match "aaa.bbb.ccc" in the list. If no match is found, it tries to match "bbb.ccc", and then "ccc". IP address matching, however, does not allow for partial matches.

Antispam filtering uses local lists for matching in the following manner:

1. **Sender IP:** The sender IP is checked against the local whitelist, then the local blacklist, and then the SBL IP-based server (if enabled).
2. **Sender Domain:** The domain name is checked against the local whitelist and then against the local blacklist.
3. **Sender E-mail Address:** The sender e-mail address is checked against the local whitelist and then against the local blacklist.

By default, the device first checks incoming e-mail against the local whitelist and blacklist. If the sender is not found on either list, the device proceeds to query the SBL server over the Internet. When both server-based antispam filtering and local list antispam filtering are enabled, checks are done in the following order:

1. The local whitelist is checked. If there is a match, no further checking is done. If there is no match...
2. The local blacklist is checked. If there is a match, no further checking is done. If there is no match...
3. The SBL server list is checked.



NOTE: Local black and whitelist matching continues after the antispam license key is expired.

Related Topics

- [Local List Antispam Filtering Configuration Overview on page 595](#)
- [Example: Configuring Local List Antispam Filtering on page 596](#)
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

Local List Antispam Filtering Configuration Overview

For each UTM feature, configure feature parameters in the following order:

1. Configure UTM custom objects for the feature:

```
user@host# set security utm custom-objects url-pattern
```

2. Configure the main feature parameters, using feature profiles.

```
user@host# set security utm feature-profile anti-spam
```

3. Configure a UTM policy for each protocol, and attach this policy to a profile.

```
user@host# set security utm utm-policy utmp1 anti-spam smtp-profile smtp1
```

4. Attach the UTM policy to a security policy.

```
user@host# set security policies from-zone trust to-zone untrust policy p1 then permit  
application-services utm-policy utmp1
```

- Related Topics**
- Understanding Local List Antispam Filtering on page 595
 - Example: Configuring Local List Antispam Filtering on page 596
 - *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Configuring Local List Antispam Filtering

This example shows how to configure local list antispam filtering.

- Requirements on page 596
- Overview on page 596
- Configuration on page 596
- Verification on page 602

Requirements

Before you begin, review how to configure the feature parameters for each UTM feature. See “Local List Antispam Filtering Configuration Overview” on page 595.

Overview

Antispam filtering uses local lists for matching. When creating your own local whitelist and blacklist for antispam filtering, you can filter against domain names, e-mail addresses, and/or IP addresses.

Configuration

- CLI Quick Configuration** To quickly configure local list antispam filtering, copy the following commands and paste them into the CLI.

```
[edit]  
set security utm custom-objects url-pattern as-black value [150.61.8.134]  
set security utm custom-objects url-pattern as-white value [150.1.2.3]  
set security utm custom-objects custom-url-category whitecusturl1 value as-white  
set security utm feature-profile anti-spam address-whitelist whitecusturl1  
set security utm feature-profile anti-spam sbl profile localprofile1  
set security utm feature-profile anti-spam sbl profile localprofile1 spam-action block  
set security utm feature-profile anti-spam sbl profile localprofile1 custom-tag-string  
***spam***  
set security utm utm-policy spampolicy2 anti-spam smtp-profile localprofile1  
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy2 match  
source-address any
```

```

set security policies from-zone trust to-zone untrust policy utmsecuritypolicy2 match
destination-address any
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy2 match
application junos-smtp
set security policies from-zone trust to-zone untrust policy utmsecuritypolicy2 then permit
application-services utm-policy spampolicy2

```

J-Web Quick Configuration

To configure local list antispam filtering:

1. Create local whitelist and blacklist custom objects by configuring a URL pattern list.
 - a. Select **Configure>Security>UTM>Custom Objects**.
 - b. In the UTM custom objects configuration window, select the **URL Pattern List** tab.
 - c. Click **Add** to create URL pattern lists.
 - d. Next to URL Pattern Name, type a unique name.



NOTE: If you are creating a whitelist, it is helpful to indicate this in the list name. The same applies to a blacklist. The name you enter here becomes available in the Address Whitelist and Address Blacklist fields when you are configuring your antispam profiles.

- e. Next to URL Pattern Value, type the URL pattern for whitelist or blacklist antispam filtering.

When entering the URL pattern, note the following wildcard character support:

- The `*\.[]\?` wildcard characters are supported.
- You must precede all wildcard URLs with **http://**.
- You can use the asterisk `*` wildcard character only if it is at the beginning of the URL and is followed by a period.
- You can use the question mark `?` wildcard character only at the end of the URL.
- The following wildcard syntax is supported: **http://*.juniper.net**, **http://www.juniper.ne?**, **http://www.juniper.n??**.
- The following wildcard syntax is not supported: ***juniper.net**, **www.juniper.ne?**, **http://*juniper.net**, **http://***.



NOTE: Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure a custom URL category list.

2. Configure a custom URL category list custom object.
 - a. Select **Configure>Security>UTM>Custom Objects**.
 - b. In the UTM custom objects configuration window, select the **URL Category List** tab.
 - c. Click **Add** to create URL category lists.
 - d. Next to URL Category Name, type a unique name. This name appears in the Address Whitelist list when you configure antispam global options.
 - e. In the Available Values box, select a URL Pattern List name from the list for bypassing scanning and move it to the Selected Values box.
3. Configure antispam filtering to use the whitelist and blacklist custom objects.
 - a. Select **Configure>Security>UTM>Global options**.
 - b. In the right pane, select the **Anti-Spam** tab.
 - c. Under Anti-Spam, select an Address Whitelist and/or an Address Blacklist from the list for local lists for spam filtering. (These lists are configured as custom objects.)
 - d. Click **OK**.
 - e. If the configuration item is saved successfully, you receive a confirmation, and you must click **OK** again. If it is not saved successfully, click **Details** in the pop-up window to discover why.
 - f. In the left pane under Security, select the **Anti-Spam** tab.
 - g. Click **Add** to configure an anti-spam profile. The profile configuration pop-up window appears.
 - h. In the Profile name box, enter a unique name.
 - i. If you are using the default server, select **Yes** beside Default SBL server. If you are not using the default server, select **No**.



NOTE: If you select No, you are disabling server-based spam filtering. You disable it only if you are using local lists or if you do not have a license for server-based spam filtering.

- j. In the Custom tag string box, type a custom string for identifying a message as spam. By default, the device uses *****SPAM*****.
 - k. In the Actions list, select the action that the device should take when it detects spam. Options include Tag subject, Block email, and Tag header.
4. Configure a UTM policy for SMTP, and attach the antispam profile to this UTM policy.
 - a. Select **Configure>Security>Policy>UTM Policies**.
 - b. In the UTM policy configuration window, click **Add** to configure a UTM policy. The policy configuration pop-up window appears.
 - c. Select the **Main** tab.
 - d. In the Policy name box, type a unique name.
 - e. In the Session per client limit box, type a session per client limit. Valid values range from 0 through 20000.
 - f. From the Session per client over limit list, select the action that the device should take when the session per client limit for this UTM policy is exceeded. Options include Log and permit and Block.
 - g. Select the **Anti-Spam profiles** tab.
 - h. From the SMTP profile list, select the antispam profile that you are attaching to this UTM policy.
5. Attach the UTM policy to a security policy.
 - a. Select **Configure>Security>Policy>FW Policies**.
 - b. In the Security Policy window, click **Add** to configure a security policy with UTM. The policy configuration pop-up window appears.
 - c. In the Policy tab, type a name in the **Policy Name** box.
 - d. Next to From Zone, select a zone from the list.
 - e. Next to To Zone, select a zone from the list.
 - f. Choose a source address.
 - g. Choose a destination address.
 - h. Choose an application by selecting **junos-smtp** (for antispam) in the Application Sets box and move it to the Matched box.
 - i. Next to Policy Action, select one of the following: **Permit**, **Deny**, or **Reject**.



NOTE: When you select Permit for policy action, several additional fields become available in the Applications Services tab, including UTM Policy.

- j. Select the **Application Services** tab.
- k. Next to UTM Policy, select the appropriate policy from the list. This attaches your UTM policy to the security policy.
- l. Click **OK** to check your configuration and save it as a candidate configuration.
- m. If the policy is saved successfully, you receive a confirmation, and you must click **OK** again. If the profile is not saved successfully, click **Details** in the pop-up window to discover why.



NOTE: You must activate your new policy to apply it.

- n. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure local list antispam filtering:

1. Configure the local list spam blocking by first creating your global local spam lists.

```
[edit security]
user@host# set utm custom-objects url-pattern as-black value [150.61.8.134]
user@host# set utm custom-objects url-pattern as-white value [150.1.2.3]
```
2. Configure a custom URL category list custom object, using the URL pattern list that you created.

```
[edit security]
user@host# set utm custom-objects custom-url-category whitecusturl1 value as-white
```
3. Configure the local list antispam feature profile by first attaching your custom-object blacklist or whitelist or both.

```
[edit security]
user@host# set utm feature-profile anti-spam address-whitelist whitecusturl1
```



NOTE: When both the whitelist and the blacklist are in use, the whitelist is checked first. If there is no match, then the blacklist is checked.

4. Configure a profile for your local list spam blocking.

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile localprofile1
```




NOTE: Although you are not using the sbl for local list spam blocking, you configure your profile from within that command similar to the server-based spam blocking procedure.

5. Configure the action to be taken by the device when spam is detected (block, tag-header, tag-subject).

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile localprofile1 spam-action
block
```

6. Configure a custom string for identifying a message as spam.

```
[edit security]
user@host# set utm feature-profile anti-spam sbl profile localprofile1
custom-tag-string ***spam***
```

7. Configure a UTM policy for SMTP to which you attach the antispam feature profile.

```
[edit security]
user@host# set utm utm-policy spampolicy2
```

8. Attach the spam feature profile to the UTM policy.

```
[edit security]
user@host# set utm utm-policy spampolicy2 anti-spam smtp-profile localprofile1
```

9. Configure a security policy for UTM, and attach the UTM policy to the security policy.

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy2 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy2 match destination-address any
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy2 match application junos-smtp
user@host# set security policies from-zone trust to-zone untrust policy
utmsecuritypolicy2 then permit application-services utm-policy spampolicy2
```

Results From configuration mode, confirm your configuration by entering the **show security utm** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security utm
custom-objects {
  anti-spam {
    url-pattern patternwhite;
    address-whitelist whitecusturl1;
    sbl {
      profile localprofile1 {
        spam-action block;
        custom-tag-string ***spam***;
      }
    }
  }
}
```

```

    }
  }
  utm-policy spampolicy2 {
    anti-spam {
      smtp-profile localprofile1;
    }
  }
}

[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy utmsecuritypolicy2 {
    match {
      source-address any;
      destination-address any;
      application junos-smtp;
    }
    then {
      permit {
        application-services {
          utm-policy spampolicy2;
        }
      }
    }
  }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Antispam Statistics on page 602

Verifying Antispam Statistics

Purpose Verify the antispam statistics.

Action From operational mode, enter the **show security utm anti-spam status** and **show security utm anti-spam statistics** commands.

The following information appears:

```

SBL Whitelist Server:
SBL Blacklist Server:
server.juniper.net
DNS Server:
Primary : 1.2.3.4, Src Interface: ge-0/0/0
Secondary: 2.3.4.5, Src Interface: ge-0/0/1
Ternary : 0.0.0.0, Src Interface: fe-0/0/2

Total connections: #
Denied connections: #
Total greetings: #
Denied greetings: #
Total e-mail scanned: #
Spam total: #

```

Spam tagged: #
Spam dropped: #
DNS errors: #
Timeout errors: #
Return errors: #
Invalid parameter errors: #
Statistics start time:
Statistics for the last 10 days.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Local List Antispam Filtering on page 595

Understanding Spam Message Handling

There are two possible actions the device can take when spam is detected. It can perform a drop action or a tag action.

- Blocking Detected Spam on page 603
- Tagging Detected Spam on page 603

Blocking Detected Spam

The device can block and drop detected spam at either the connection level or the e-mail level:

- Blocking spam at the connection level

When the SMTP sender is identified as a spam sender based on its IP address, the SMTP connection is rejected and dropped. An error message with a proper error code from the firewall is sent out on behalf of the SMTP server. An example of such an error message is:

554 Transaction failed due to anti spam setting

- Blocking spam at the e-mail level

When a particular e-mail sender is identified as spam sender based on its sender address, the e-mail is rejected and dropped. An error message with a proper error code from the firewall is sent back to the sender on behalf of the server. An example of such an error message is:

550 Requested action not taken: mailbox unavailable

Tagging Detected Spam

The device can allow and tag the e-mail if the message sender is detected as a spammer. This tagging can occur at the connection level so that all the e-mails for the connection in question are tagged. Otherwise, you can tag only an individual e-mail. Two tagging methods are supported:

- Tag the subject: A user-defined string is added at the beginning of the subject of the e-mail.

- Tag the header: A user-defined string is added to the e-mail header.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

CHAPTER 31

Full Antivirus Protection

- Full Antivirus Protection Overview on page 605
- Full Antivirus Scanner Pattern Database on page 606
- Full Antivirus File Scanning on page 609
- Full Antivirus Application Protocol Scanning on page 615
- Full Antivirus Scan Results and Notification Options on page 626
- Full Antivirus Configuration Overview on page 631
- Configuring Full Antivirus (J-Web Procedure) on page 632
- Example: Configuring Full Antivirus (CLI) on page 638
- Monitoring Antivirus Sessions and Scan Results on page 644

Full Antivirus Protection Overview

A virus is executable code that infects or attaches itself to other executable code in order to reproduce itself. Some malicious viruses erase files or lock up systems, while other viruses merely infect files and can overwhelm the target host or network with bogus data. The full file-based antivirus feature provides file-based scanning on specific Application Layer traffic checking for viruses against a virus signature database. It collects the received data packets until it has reconstructed the original application content, such as an e-mail file attachment, and then scans this content.

The full file-based antivirus scanning feature is a separately licensed subscription service. Kaspersky Lab provides the scan engine for full file-based antivirus. When your antivirus license key expires, you can continue to use locally stored antivirus signatures without any updates. But in that case, if the local database is deleted, antivirus scanning is disabled.



NOTE: The express antivirus feature provides better performance but lower security. Note that if you switch from full file-based antivirus protection to express antivirus protection, you must reboot the device in order for express antivirus to begin working.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Scanner Pattern Database

- Understanding Full Antivirus Pattern Updates on page 606
- Full Antivirus Pattern Update Configuration Overview on page 607
- Example: Specifying the Full Antivirus Pattern Update Server (CLI) on page 607
- Example: Automatically Updating Full Antivirus Patterns (J-Web) on page 608
- Example: Automatically Updating Full Antivirus Patterns (CLI) on page 608
- Manually Updating, Reloading, and Deleting Full Antivirus Patterns (CLI Procedure) on page 608

Understanding Full Antivirus Pattern Updates

The full file-based antivirus protection signature database is called the Juniper Full antivirus database (downloaded by the **pattern-update** command). This database is different from the database used by express antivirus. It detects all destructive malicious code, including viruses (polymorphic and other advanced virus types), worms, Trojans, and malware.

Updates to the pattern file are added as new viruses are discovered. When Kaspersky Lab updates the signatures in its pattern database, the security device downloads these updates so that the antivirus scanner is using the latest, most up-to-date signatures when scanning traffic. The security device can perform these updates automatically (the default), or you can perform pattern update downloads manually.

The database pattern server is accessible through HTTP or HTTPS. By default, the antivirus module checks for database updates automatically every 60 minutes. You can change this interval and you can trigger updates manually, as well. The number of files that are downloaded during an update and the duration of the download process can vary.

A local copy of the pattern database is saved in persistent data storage (that is, the flash disk). If the device is rebooted, the local copy remains available for the antivirus scan engine to use during the antivirus scan engine initialization time, without the need for network access to the pattern database server.



NOTE: If the auto-update fails, the updater automatically retries to update three more times. If the database download continues to fail, the updater stops trying and waits for the next periodic update before trying again.



NOTE: Once your subscription expires, you have a 30 day grace period during which you can continue to update the antivirus pattern file. Once that grace period expires, the update server no longer permits antivirus pattern file updates.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Pattern Update Configuration Overview

Before you begin, there are several prerequisites that must be met in order to perform a successful pattern database update:

- You must have a valid antivirus scanner license.
- You must have network connectivity and access to the pattern database server.
- Your DNS settings and port settings (port 80) must be correct.

To update the patterns for the antivirus signature database:

1. On the security device, specify the URL address of the pattern-update server.
2. (Optional) Specify how often the device should automatically check for pattern-server updates.

After the security device downloads the server-initialization file, the device checks that the pattern file is valid. The device then parses the file to obtain information about it, including the file version, size, and location of the pattern file server.

If the pattern file on the security device is out-of-date (or nonexistent because this is the first time you are loading it), and, if the antivirus pattern-update service subscription is still valid, the device automatically retrieves an updated pattern file from the pattern file server.

The following is an example of the CLI for configuring the database update feature:

```
utm {
  feature-profile {
    anti-virus {
      type
      kaspersky-lab-engine {
        pattern-update
        url url
        interval minutes
      }
    }
  }
}
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Specifying the Full Antivirus Pattern Update Server (CLI)

To specify the pattern-update server on the security device, enter the URL address of the pattern-update server. In this example, you update the URL for an SRX210 Services Gateway.

To specify the pattern-update server, enter the following CLI statement:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine
pattern-update url http://update.juniper-updates.net/AV/SRX210
```

By default, the Juniper-Kaspersky URL for full antivirus is:

<http://update.juniper-updates.net/AV/SRX210>, where SRX210 is the name of your hardware device. This part of the URL is different and platform specific for each platform.



NOTE: Other than the platform name, you should not change this URL unless you are experiencing problems with it and have called for support.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Automatically Updating Full Antivirus Patterns (J-Web)

In this example, you configure the security device to update the pattern file automatically every 120 minutes. (The default antivirus pattern-update interval is 60 minutes.)

To automatically update antivirus patterns:

1. Select **Configure>UTM>Anti-Virus**.
2. Next to Interval, in the Kaspersky Lab Engine section, enter **120** in the box.
3. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Automatically Updating Full Antivirus Patterns (CLI)

In this example, you configure the security device to update the pattern file automatically every 120 minutes. (The default antivirus pattern-update interval is 60 minutes.)

To automatically update antivirus patterns, enter the following CLI statement:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine
pattern-update interval 120
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Manually Updating, Reloading, and Deleting Full Antivirus Patterns (CLI Procedure)

To manually update antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus kaspersky-lab-engine pattern-update
```

To manually reload antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus kaspersky-lab-engine pattern-reload
```

To manually delete antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus kaspersky-lab-engine pattern-delete
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus File Scanning

- Understanding the Full Antivirus Internal Scan Engine on page 609
- Global, Profile-Based, and Policy-Based Full Antivirus Scan Settings on page 609
- Full Antivirus Scan Modes on page 611
- Full Antivirus Intelligent Prescreening on page 612
- Full Antivirus Content Size Limits on page 612
- Full Antivirus Decompression Layer Limit on page 613
- Full Antivirus Scanning Timeout on page 614
- Full Antivirus Scan Session Throttling on page 615

Understanding the Full Antivirus Internal Scan Engine

The full file-based antivirus module is the software subsystem on the gateway device that scans specific Application Layer traffic to protect users from virus attacks and to prevent viruses from spreading. The antivirus software subsystem consists of a virus signature database, an application proxy, the scan manager, and the scan engine.

Kaspersky Lab provides the scan engine and it works in the following manner:

1. A client establishes a TCP connection with a server and then starts a transaction.
2. If the application protocol in question is marked for antivirus scanning, the traffic is forwarded to an application proxy for parsing.
3. When the scan request is sent, the scan engine scans the data by querying a virus pattern database.
4. The scan manager monitors antivirus scanning sessions, checking the properties of the data content against the existing antivirus settings.
5. After scanning has occurred, the result is then handled by the scan manager.

The Kaspersky Lab scan engine supports regular file scanning and script file scanning. With regular file scanning, the input object is a regular file. The engine matches the input content with all possible signatures. With script file scanning, the input object is a script file. It can be JavaScript, VBScript, mIRC script, bat scripts (DOS bat files), and other text scripts. The engine matches the input content only with signatures for script files. Script scanning is only applicable for HTML content over the HTTP protocol. There are two criteria for this scan type. First, the content-type field of this HTML document must be text or HTML. Second, there is no content encoding in the HTTP header. If those two criteria are met, an HTML parser is used to parse the HTML document for scripts.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Global, Profile-Based, and Policy-Based Full Antivirus Scan Settings

- Understanding Full Antivirus Scan Level Settings on page 610
- Example: Configuring Full Antivirus Scan Settings at Different Levels (CLI) on page 610

Understanding Full Antivirus Scan Level Settings

The antivirus module allows you to configure scanning options on a global level, on a UTM profile level, or on a firewall policy level. Each configuration level has the following implications:

- Global antivirus settings—Settings are applied to all antivirus sessions. Global settings are general overall configurations for the antivirus module or settings that are not specific for profiles.
- Profile-based settings—Antivirus settings are different for different protocols within the same policy.
- Policy-based settings—Antivirus settings are different for different policies. Policy-based antivirus settings are applied to all scan-specified traffic defined in a firewall policy.

The majority of antivirus settings are configured within an antivirus profile, bound to specified protocols, and used by designated policies. These UTM policies are then applied to the traffic according to firewall policies. If a firewall policy with an antivirus setting matches the properties of a traffic flow, the antivirus setting is applied to the traffic session. Therefore, you can apply different antivirus settings for different protocols and for different traffic sessions.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Configuring Full Antivirus Scan Settings at Different Levels (CLI)

To configure scanning options at different levels, enter the following CLI configuration statements.

- In this example, scanning options are configured at the global level:

```
user@host set security utm feature-profile anti-virus kaspersky-lab-engine
```

- In this example, scanning options are configured at the UTM profile level using the `kasprof1` UTM profile:

```
user@host set security utm feature-profile anti-virus kaspersky-lab-engine profile kasprof1
```

- In this example, scanning options are configured at the UTM policy level using the `p1` UTM policy:

```
user@host set security utm utm-policy p1
```

The following is an example of different antivirus settings applied to different protocols configured as profiles within a designated UTM policy:

```
edit security utm
  utm-policy name {
    anti-virus {
      http-profile av-profile
      ftp {
        upload-profile av-profile
        download-profile av-profile
      }
      smtp-profile av-profile
```

```

        pop3-profile av-profile
        imap-profile av-profile
    }
}

```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Scan Modes

- Understanding Full Antivirus Scan Mode Support on page 611
- Configuring Full Antivirus File Extension Scanning (CLI Procedure) on page 611

Understanding Full Antivirus Scan Mode Support

The Kaspersky Lab scan engine supports two modes of scanning:

- **scan-all**—This option tells the scan engine to scan all the data it receives.
- **scan-by-extension**—This option bases all scanning decisions on the file extensions found in the traffic in question.

When scanning content, you can use a file extension list to define a set of file extensions that are used in file extension scan mode (**scan-by-extension**). The antivirus module can then scan files with extensions on the scan-extension list. If an extension is not defined in an extension list, the file with that extension is not scanned in scan-by-extension mode. If there is no extension present, the file in question is scanned.

When using a file extension list to scan content, please note the following requirements:

- File extension entries are case-insensitive.
- The maximum length of the file extension list name is 29 bytes.
- The maximum length of each file extension entry is 15 bytes.
- The maximum entry number in a file extension list is 255.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Configuring Full Antivirus File Extension Scanning (CLI Procedure)

To configure file-extension scanning, use the following CLI configuration statements:

```

security utm {
  custom-objects {
    filename-extension { ; set of list
      name extension-list-name; #mandatory
      value windows-extension-string;
    }
  }
}

security utm feature-profile anti-virus kaspersky-lab-engine profile name {
  scan-options {
    scan-extension ext-list
  }
}

```

```
}
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Intelligent Prescreening

- Understanding Full Antivirus Intelligent Prescreening on page 612
- Example: Configuring Full Antivirus Intelligent Prescreening (CLI) on page 612

Understanding Full Antivirus Intelligent Prescreening

By default, intelligent prescreening is enabled to improve antivirus scanning performance. The antivirus module generally begins to scan data after the gateway device has received all the packets of a file. Intelligent prescreening tells the antivirus module to begin scanning a file much earlier. In this case, the scan engine uses the first packet or the first several packets to determine if a file could possibly contain malicious code. The scan engine does a quick check on these first packets and if it finds that it is unlikely that the file is infected, it then decides that it is safe to bypass the normal scanning procedure.



NOTE: Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for MIME encoded traffic, mail protocols (SMTP, POP3, IMAP) and HTTP POST.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Configuring Full Antivirus Intelligent Prescreening (CLI)

To configure intelligent prescreening, enter the following CLI configuration statements.

- In this example, intelligent prescreening is enabled for the kasprof1 profile:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1  
scan-options intelligent-prescreening
```
- In this example, intelligent prescreening is disabled for the kasprof1 profile:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1  
scan-options no-intelligent-prescreening
```



NOTE: Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for mail protocols (SMTP, POP3, IMAP) and HTTP POST.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Content Size Limits

- Understanding Full Antivirus Content Size Limits on page 613
- Configuring Full Antivirus Content Size Limits (CLI Procedure) on page 613

Understanding Full Antivirus Content Size Limits

Due to resource constraints, there is a default, device-dependent limit on maximum content size for the database. The content size value is configurable. There is also a lower and upper limit for maximum content size. (This range is device dependent and is not configurable.)

The content size check occurs before the scan request is sent. The exact timing of this is protocol dependent. If the protocol header contains an accurate content length field, the content size check takes place when the content length field is extracted during header parsing. The content size usually refers to file size. If there is no content length field, the size is checked while the antivirus module is receiving packets. The content size, in this case, refers to accumulated TCP payload size.



NOTE: This setting can be used in all protocols.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Configuring Full Antivirus Content Size Limits (CLI Procedure)

To configure content size limits, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine profile name {
  scan-options {
    content-size-limit KB;
  }
}
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Decompression Layer Limit

- Understanding Full Antivirus Decompression Layer Limits on page 613
- Configuring Full Antivirus Decompression Layer Limits (CLI Procedure) on page 614

Understanding Full Antivirus Decompression Layer Limits

The decompression layer limit specifies how many layers of nested compressed files and files with internal extractable objects, such as archive files (tar), MS Word and PowerPoint files, the internal antivirus scanner can decompress before it executes the virus scan. For example, if a message contains a compressed .zip file that contains another compressed .zip file, there are two compression layers. Decompressing both files requires a decompress layer setting of 2.

It is worth noting that during the transfer of data, some protocols use content encoding. The antivirus scan engine must decode this layer, which is considered a decompression level, before it scans for viruses.

There are three kinds of compressed data:

- compressed file (zip, rar, gzip)

- encoded data (MIME)
- packaged data (OLE, .CAP, .MSI, .TAR, .EML)

A decompression Layer could be a layer of a zipped file or an embedded object in packaged data. The antivirus engine scans each layer before unpacking the next layer, until it either reaches the user-configured decompress limit, reaches the device decompress layer limit, finds a virus or other malware, or decompresses the data completely, whichever comes first.

As the virus signature database becomes larger and the scan algorithms become more sophisticated, the scan engine has the ability to look deeper into the data for embedded malware. As a result, it can uncover more layers of compressed data. The Juniper device's level of security is limited by decompress limit, which is based on the memory allocated to the security service. If a virus is not found within the decompress limit, the user has an option to either pass or drop the data.



NOTE: This setting can be used in all protocols.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Configuring Full Antivirus Decompression Layer Limits (CLI Procedure)

To configure decompression layer limits, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine profile name {  
  scan-options {  
    decompress-layer-limit number  
  }  
}
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Scanning Timeout

- Understanding Full Antivirus Scanning Timeouts on page 614
- Configuring Full Antivirus Scanning Timeouts (CLI Procedure) on page 615

Understanding Full Antivirus Scanning Timeouts

The scanning timeout value includes the time frame from when the scan request is generated to when the scan result is returned by the scan engine. The time range can be 1 to 1800 seconds. By default, it is 180 seconds.



NOTE: This timeout parameter is used by all supported protocols. Each protocol can have a different timeout value.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Configuring Full Antivirus Scanning Timeouts (CLI Procedure)

To configure scanning timeouts, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine profile name {
  scan-options {
    timeout-value seconds {
    }
  }
}
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Scan Session Throttling

- Understanding Full Antivirus Scan Session Throttling on page 615
- Configuring Full Antivirus Scan Session Throttling (CLI Procedure) on page 615

Understanding Full Antivirus Scan Session Throttling

In an attempt to consume all available resources and hinder the ability of the scan engine to scan other traffic, a malicious user might generate a large amount of traffic all at once. To prevent such activity from succeeding, a session throttle is imposed for antivirus resources, thereby restricting the amount of traffic a single source can consume at one time. The limit is an integer with 100 as the default setting. This integer refers to the maximum allowed sessions from a single source. You may change this default limit, but understand that if this limit is set high, that is comparable to no limit.

Over-limit is a fallback setting for the connection-per-client limit. The default behavior of over-limit is to block sessions. This is a per-policy setting. You can specify different settings for different UTM policies.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Configuring Full Antivirus Scan Session Throttling (CLI Procedure)

To configure scan session throttling, use the following CLI configuration statements:

```
security utm utm-policy name
  traffic-options {
    sessions-per-client {
      limit number;
      over-limit { log-and-permit | block }
    }
  }
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Application Protocol Scanning

- Understanding Full Antivirus Application Protocol Scanning on page 616
- HTTP Full Antivirus Scanning on page 617
- FTP Full Antivirus Scanning on page 620

- SMTP Full Antivirus Scanning on page 621
- POP3 Full Antivirus Scanning on page 622
- IMAP Full Antivirus Scanning on page 624

Understanding Full Antivirus Application Protocol Scanning

You can turn antivirus scanning on and off on a per protocol basis. If scanning for a protocol is disabled in an antivirus profile, there is no application intelligence for this protocol. Therefore, in most cases, traffic using this protocol is not scanned. But if the protocol in question is based on another protocol for which scanning is enabled in an antivirus profile, then the traffic is scanned as that enabled protocol.

The internal antivirus scan engine supports scanning for specific Application Layer transactions allowing you to select the content (HTTP, FTP, SMTP, POP3, or IMAP traffic) to scan. For each content type that you are scanning, you have different configuration options.

Profile-based settings, including enable/disable, scan-mode, and scan result handling settings, may not be applicable to all supported protocols. The following table lists profile-based settings and their protocol support.

Table 68: Supported Profile-based Settings By Protocol

Profile Setting	Protocol Support
Enable or disable scanning on per protocol basis	All protocols support this feature
"Full Antivirus Scan Modes" on page 611, including file extension scanning	All protocols support this feature
"Full Antivirus Content Size Limits" on page 612	All protocols support this feature
"Full Antivirus Decompression Layer Limit" on page 613	All protocols support this feature
"Full Antivirus Scanning Timeout" on page 614	All protocols support this feature
"Understanding HTTP Tricking" on page 618	HTTP only
"Understanding Antivirus Scanning Fallback Options" on page 629	All protocols support this feature
"Protocol-Only Virus-Detected Notifications" on page 627	All protocols support this feature
"E-Mail Virus-Detected Notifications" on page 627	SMTP, POP3, and IMAP only
"Custom Message Virus-Detected Notifications" on page 628	All protocols support this feature

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

HTTP Full Antivirus Scanning

- Understanding HTTP Scanning on page 617
- Enabling HTTP Scanning (CLI Procedure) on page 618
- Understanding HTTP Trickling on page 618
- Configuring HTTP Trickling to Prevent Timeouts During Antivirus Scanning (CLI Procedure) on page 618
- Understanding MIME Whitelists on page 618
- Example: Configuring MIME Whitelists to Bypass Antivirus Scanning (CLI) on page 619
- Understanding URL Whitelists on page 619
- Configuring URL Whitelists to Bypass Antivirus Scanning (CLI Procedure) on page 620

Understanding HTTP Scanning

If antivirus scanning is enabled for Hypertext Transfer Protocol (HTTP) traffic in a content security profile, TCP traffic to defined HTTP service ports (generally port 80) is monitored. For HTTP traffic, the security device scans both HTTP responses and requests (get, post, and put commands).



NOTE: For HTTP antivirus scanning, both HTTP 1.0 and 1.1 are supported. If the protocol version is HTTP 0.x, the antivirus scanner attempts to scan the traffic. Unknown protocols are bypassed. For example, some application protocols use HTTP as the transport but do not comply with HTTP 1.0 or 1.1. These are considered unknown protocols and are not scanned.

This is a general description of how HTTP traffic is intercepted, scanned, and acted upon by the antivirus scanner:

1. An HTTP client sends an HTTP request to a webserver or a webserver responds to an HTTP request.
2. The security device intercepts the request and passes the data to the antivirus scanner, which scans it for viruses.
3. After completing the scan, the device follows one of two courses:
 - If there is no virus, the device forwards the request to the webserver.
 - If there is a virus, the device drops the request and sends an HTTP message reporting the infection to the client.

With script-only scanning, the input object is a script file. It can be JavaScript, VBScript, mIRC script, bat scripts (DOS bat files) and other text scripts. The engine matches the input content only with signatures for script files. Script scanning is applicable only for HTML content over the HTTP protocol. There are two criteria for this scan-type. First, the content-type field of this HTML document must be text or HTML. Second, there is no content encoding in the HTTP header. If those two criteria are met, an HTML parser is used to parse the HTML document.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Enabling HTTP Scanning (CLI Procedure)

To enable antivirus scanning for HTTP traffic, enter the following CLI configuration statement:

```
user@host# set security utm utm-policy policy-name anti-virus http
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Understanding HTTP Trickling

HTTP trickling is a mechanism used to prevent the HTTP client or server from timing-out during a file transfer or during antivirus scanning. On some slow link transferring, a large file could timeout if too much time is taken for the antivirus scanner to scan a complex file.

HTTP trickling is the forwarding of specified amounts of unscanned HTTP traffic to the requesting HTTP client to prevent the browser window from timing out while the scan manager examines downloaded HTTP files. (The security device forwards small amounts of data in advance of transferring an entire scanned file.)

HTTP Trickling is time-based and there is only one parameter, the time-out interval, to configure for this feature. By default, trickling is disabled.



NOTE: The timeout based trickling is packet driven. This means, if no packet is received within a certain time frame, HTTP trickling is discontinued. This setting is only supported for HTTP connections.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Configuring HTTP Trickling to Prevent Timeouts During Antivirus Scanning (CLI Procedure)

To configure HTTP trickling, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine {  
  profile name {  
    trickling timeout seconds;  
  }  
}
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Understanding MIME Whitelists

The gateway device uses MIME (Multipurpose Internet Mail Extension) types to decide which traffic may bypass antivirus scanning. The MIME whitelist defines a list of MIME types and can contain one or many MIME entries.

A MIME entry is case-insensitive. An empty MIME is an invalid entry and should never appear in the MIME list. If the MIME entry ends with a / character, prefix matching takes place. Otherwise, exact matching occurs.

There are two types of MIME lists used to configure MIME type antivirus scan bypassing:

- **mime-whitelist list**—This is the comprehensive list for those MIME types that can bypass antivirus scanning.
- **exception list**—The exception list is a list for excluding some MIME types from the mime-whitelist list. This list is a subset of MIME types found in the mime-whitelist.

For example, if the mime-whitelist includes the entry, **video/** and the exception list includes the entry **video/x-shockwave-flash**, by using these two lists, you can bypass objects with “video/” MIME type but not bypass “video/x-shockwave-flash” MIME type.

You should note that there are limits for mime-whitelist entries as follows:

- The maximum number of MIME items in a MIME list is 50.
- The maximum length of each MIME entry is restricted to 40 bytes.
- The maximum length of a MIME list name string is restricted to 40 bytes.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Configuring MIME Whitelists to Bypass Antivirus Scanning (CLI)

To configure MIME whitelists:

1. Create a MIME whitelist and a MIME exception list for antivirus scanning. In this example, you create the `avmime1` and `ex-avmime1` lists:

```
user@host# set security utm custom-objects mime-pattern avmime1
user@host# set security utm custom-objects mime-pattern ex-avmime1
```

2. Add MIME patterns to the lists:

```
user@host# set security utm custom-objects mime-pattern avmime1 value
[video/quicktime image/x-portable-anymap x-world/x-vrml]
user@host# set security utm custom-objects mime-pattern ex-avmime1 value
[video/quicktime-inappropriate]
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Understanding URL Whitelists

A URL whitelist is a unique custom list that you define in which all the URLs or IP addresses in that list for a specified category are always bypassed for scanning. Because antivirus scanning is CPU and memory intensive action, if there are URLs or IP addresses that you are sure do not require scanning, you may want to create this custom list add them to it.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Configuring URL Whitelists to Bypass Antivirus Scanning (CLI Procedure)

To configure URL whitelists, use the following CLI configuration statements:

```
security utm custom-objects {  
  custom-url-category { ; set of list  
    name url-category-name; #mandatory  
    value url-pattern-name;  
  }  
}
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

FTP Full Antivirus Scanning

- Understanding FTP Antivirus Scanning on page 620
- Enabling FTP Antivirus Scanning (CLI Procedure) on page 620

Understanding FTP Antivirus Scanning

If antivirus scanning is enabled for File Transfer Protocol (FTP) traffic in a content security profile, the security device monitors the control channel and, when it detects one of the FTP commands for transferring data, it scans the data sent over the data channel.

This is a general description of how FTP traffic is intercepted, scanned, and acted upon by the antivirus scanner:

1. A local FTP client opens an FTP control channel to an FTP server and requests the transfer of some data.
2. The FTP client and server negotiate a data channel over which the server sends the requested data. The security device intercepts the data and passes it to the antivirus scan engine, which scans it for viruses.
3. After completing the scan, the device follows one of two courses:
 - If there is no virus, the device forwards the data to the client.
 - If there is a virus, the device replaces the data with a drop message in the data channel and sends a message reporting the infection in the control channel.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Enabling FTP Antivirus Scanning (CLI Procedure)

To enable antivirus scanning for File Transfer Protocol (FTP) traffic, enter the following CLI configuration statement:

```
user@host# security utm utm-policy policy-name anti-virus ftp
```



NOTE: In order to scan FTP traffic, the FTP ALG must be enabled.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

SMTP Full Antivirus Scanning

- Understanding SMTP Antivirus Scanning on page 621
- Enabling SMTP Antivirus Scanning (CLI Procedure) on page 622

Understanding SMTP Antivirus Scanning

If SMTP (Simple Mail Transfer Protocol) antivirus scanning is enabled in a content security profile, the security device redirects traffic from local SMTP clients to the antivirus scanner before sending it to the local mail server.



NOTE: Chunking is an alternative to the data command. It provides a mechanism to transmit a large message in small chunks. It is not supported. Messages using chunking are bypassed and are not scanned.

This is a general description of how SMTP traffic is intercepted, scanned, and acted upon by the antivirus scanner:

1. An SMTP client sends an e-mail message to a local mail server or a remote mail server forwards an e-mail message via SMTP to the local mail server.
2. The security device intercepts the e-mail message and passes the data to the antivirus scanner, which scans it for viruses.
3. After completing the scan, the device follows one of two courses:
 - If there is no virus, the device forwards the message to the local server.
 - If there is a virus, the device sends a replacement message to the client.

This topic includes the following sections:

- Understanding SMTP Antivirus Mail Message Replacement on page 621
- Understanding SMTP Antivirus Sender Notification on page 622
- Understanding SMTP Antivirus Subject Tagging on page 622

Understanding SMTP Antivirus Mail Message Replacement

If the antivirus scanner finds a virus in an e-mail message, the original message is dropped, the message body is truncated, and the content is replaced by a message that may appear as follows:

```
nContent-Type: text/plain
Your mail <src_ip> : <src_port> — <dst_port>: <dst_port> contains contaminated file
<filename> with virus <virusname>, so it is dropped.
```

If a scan error is returned and the fail mode is set to drop, the original message is dropped and the entire message body is truncated. The content is replaced by a message that may appear as follows:

```
nContent-Type: text/plain
Your mail <src_ip> : <src_port> — <dst_port>: <dst_port> is dropped for <reason>.
```

Understanding SMTP Antivirus Sender Notification

If **notify-sender-on-virus** is set and the message is dropped due to a detected virus, an e-mail is sent to the mail sender. The content of the notification may appear as follows:

```
From: <admin>@<gateway_ip>
To: <sender_e-mail>
Subject: Mail Delivery Failure
This message is created automatically by mail delivery software. A message that you sent
could not be delivered to one or more of its recipients for the reason:
<src_ip> : <src_port> — <dst_port>: <dst_port> <ENVID> contaminated file <filename>
with virus <virusname>.
e-mail Header is:
<header of scanned e-mail>
```

If **notify-sender-on-error-drop** is set and the message is dropped due to a scan error, an e-mail is sent to the mail sender of the scanned message. The content of the e-mail may appear as follows:

```
From: <admin>@<gateway_ip>
To: <sender_e-mail>
Subject: Mail Delivery Failure
This message is created automatically by mail delivery software. A message that you sent
could not be delivered to one or more of its recipients for the reason:
<src_ip> : <src_port> — <dst_port>: <dst_port> <ENVID> <reason>.
e-mail Header is:
<header of scanned e-mail>
```



NOTE: For information on the ENVID parameter, refer to RFC 3461.

Understanding SMTP Antivirus Subject Tagging

If a scan error is returned and the fail mode is set to **pass**, the antivirus module passes the message through to the server. If **notify-recipient-on-error-pass** is set, the following string is appended to the end of the subject field:

(No virus check: <reason>)

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Enabling SMTP Antivirus Scanning (CLI Procedure)

To enable antivirus scanning for SMTP traffic, enter the following CLI configuration statement:

```
user@host# set security utm utm-policy policy-name anti-virus smtp-profile
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

POP3 Full Antivirus Scanning

- Understanding POP3 Antivirus Scanning on page 623
- Enabling POP3 Antivirus Scanning (CLI Procedure) on page 624

Understanding POP3 Antivirus Scanning

If Post Office Protocol 3 (POP3) antivirus scanning is enabled in a content security profile, the security device redirects traffic from a local mail server to antivirus scanner before sending it to the local POP3 client.

This is a general description of how POP3 traffic is intercepted, scanned, and acted upon by the antivirus scanner.

1. The POP3 client downloads an e-mail message from the local mail server.
2. The security device intercepts the e-mail message and passes the data to the antivirus scanner, which scans it for viruses.
3. After completing the scan, the security device follows one of two courses:
 - If there is no virus, the device forwards the message to the client.
 - If there is a virus, the device sends a message reporting the infection to the client.



NOTE: See “Protocol-Only Virus-Detected Notifications” on page 627 for information on protocol-only notifications for IMAP.

This topic includes the following sections:

- Understanding POP3 Antivirus Mail Message Replacement on page 623
- Understanding POP3 Antivirus Sender Notification on page 623
- Understanding POP3 Antivirus Subject Tagging on page 624

Understanding POP3 Antivirus Mail Message Replacement

If the antivirus scanner finds a virus in an e-mail message, the original message is dropped, the message body is truncated, and the content is replaced by a message that may appear as follows:

```
nContent-Type: text/plain
Your mail <src_ip> : <src_port> — <dst_port>: <dst_port> contains contaminated file
<filename> with virus <virusname>, so it is dropped.
```

Understanding POP3 Antivirus Sender Notification

If **notify-sender-on-virus** is set and the message is dropped due to a detected virus, an e-mail is sent to the mail sender.

```
From: <admin>@<gateway_ip>
To: <sender_e-mail>
Subject: Mail Delivery Failure
This message is created automatically by mail delivery software. A message that you sent
could not be delivered to one or more of its recipients for the reason:
<src_ip> : <src_port> — <dst_port>: <dst_port> contaminated file <filename> with virus
<virusname>.
e-mail Header is:
<header of scanned e-mail>
```

If **notify-sender-on-error-drop** is set and the message is dropped due to a scan error, an e-mail is sent to the mail sender of the scanned message. The content of the e-mail may appear as follows:

```
From: <admin>@<gateway_ip>
To: <sender_e-mail>
Subject: Mail Delivery Failure
This message is created automatically by mail delivery software. A message that you sent
could not be delivered to one or more of its recipients for the reason:
<src_ip> : <src_port> — <dst_port>: <dst_port> <reason>.
e-mail Header is:
<header of scanned e-mail>
```

Understanding POP3 Antivirus Subject Tagging

If a scan error is returned and the fail mode is set to **pass**, the antivirus module passes the message through to the server. If **notify-recipient-on-error-pass** is set, the following string is appended to the end of subject field:

```
(No virus check: <reason>)
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Enabling POP3 Antivirus Scanning (CLI Procedure)

To enable antivirus scanning for POP3 traffic, enter the following CLI configuration statement:

```
user@host# set security utm utm-policy policy-name anti-virus pop3-profile
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

IMAP Full Antivirus Scanning

- Understanding IMAP Antivirus Scanning on page 624
- Enabling IMAP Antivirus Scanning (CLI Procedure) on page 626

Understanding IMAP Antivirus Scanning

If IMAP (Internet Message Access Protocol) antivirus scanning is enabled in a content security profile, the security device redirects traffic from a local mail server to the internal antivirus scanner before sending it to the local IMAP client.

This is a general description of how IMAP traffic is intercepted, scanned, and acted upon by the antivirus scanner.

1. The IMAP client downloads an e-mail message from the local mail server.
2. The security device intercepts the e-mail message and passes the data to the antivirus scanner, which scans it for viruses.
3. After completing the scan, the security device follows one of two courses:
 - If there is no virus, the device forwards the message to the client.
 - If there is a virus, the device sends a message reporting the infection to the client.



NOTE: See “Protocol-Only Virus-Detected Notifications” on page 627 for information on protocol-only notifications for IMAP.

This topic includes the following sections:

- Understanding IMAP Antivirus Mail Message Replacement on page 625
- Understanding IMAP Antivirus Sender Notification on page 625
- Understanding IMAP Antivirus Subject Tagging on page 626
- Understanding IMAP Antivirus Scanning Limitations on page 626

Understanding IMAP Antivirus Mail Message Replacement

If the antivirus scanner finds a virus in an e-mail message, the original message is dropped, the message body is truncated, and the content is replaced by a message that may appear as follows:

nContent-Type: text/plain

Your mail <src_ip> : <src_port> — <dst_port>: <dst_port> contains contaminated file <filename> with virus <virusname>, so it is dropped.

Understanding IMAP Antivirus Sender Notification

If **notify-sender-on-virus** is set and the message is dropped due to a detected virus, an e-mail is sent to the mail sender.

From: <admin>@<gateway_ip>

To: <sender_e-mail>

Subject: Mail Delivery Failure

This message is created automatically by mail delivery software. A message that you sent could not be delivered to one or more of its recipients for the reason:

<src_ip> : <src_port> — <dst_port>: <dst_port> contaminated file <filename> with virus <virusname>.

e-mail Header is:

<header of scanned e-mail>

If **notify-sender-on-error-drop** is set and the message is dropped due to a scan error, an e-mail is sent to the mail sender of the scanned message. The content of the e-mail may appear as follows:

From: <admin>@<gateway_ip>

To: <sender_e-mail>

Subject: Mail Delivery Failure

This message is created automatically by mail delivery software. A message that you sent could not be delivered to one or more of its recipients for the reason:

<src_ip> : <src_port> — <dst_port>: <dst_port> <reason>.

e-mail Header is:

<header of scanned e-mail>

Understanding IMAP Antivirus Subject Tagging

If a scan error is returned and the fail mode is set to **pass**, the antivirus module passes the message through to the server. If **notify-recipient-on-error-pass** is set, the following string is appended to the end of subject field:

(No virus check: <reason>)

Understanding IMAP Antivirus Scanning Limitations

Mail Fragments — It is possible to chop one e-mail into multiple parts and to send each part through a different response. This is called mail fragmenting and most popular mail clients support it in order to send and receive large e-mails. Scanning of mail fragments is not supported by the antivirus scanner and in such cases, the message body is not scanned.

Partial Content — Some mail clients treat e-mail of different sizes differently. For example, small e-mails (less than 10 KB) are downloaded as a whole. Large e-mails (e.g. less than 1 MB) are chopped into 10 KB pieces upon request from the IMAP server. Scanning of any partial content requests is not supported by the antivirus scanner.

IMAP Uploads — Only antivirus scanning of IMAP downloads is supported. IMAP upload traffic is not scanned.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Enabling IMAP Antivirus Scanning (CLI Procedure)

To enable antivirus scanning for IMAP traffic, enter the following CLI configuration statement:

```
user@host# security utm utm-policy policy-name anti-virus imap-profile
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Scan Results and Notification Options

- Understanding Full Antivirus Scan Result Handling on page 626
- Protocol-Only Virus-Detected Notifications on page 627
- E-Mail Virus-Detected Notifications on page 627
- Custom Message Virus-Detected Notifications on page 628
- Full Antivirus Scanning Fallback Options on page 629

Understanding Full Antivirus Scan Result Handling

Different antivirus scan results are handled in different manners. For example, if a scan result is clean, the traffic is forwarded to the receiver. If the scan result is infected, the traffic is dropped. If the scan results in an error, the result handling depends on the cause of the failure and the configuration (fallback settings).

The following is a list of actions based on scan results:

- Scan Result = Pass

The scan result handling action is to pass the message. In this case, no virus is detected and no error code is returned. Or, an error code is returned, but the fallback option for this error code is set to log-and-permit.

- Scan Result = Block

The scan result handling action is to block the message. In this case, either a virus is detected or an error code is returned and the fallback option for this error code is BLOCK.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Protocol-Only Virus-Detected Notifications

- Understanding Protocol-Only Virus-Detected Notifications on page 627
- Configuring Protocol-Only Virus-Detected Notifications (CLI Procedure) on page 627

Understanding Protocol-Only Virus-Detected Notifications

When content is blocked because a virus is found or a scan error occurs, the client generally still receives a successful response code but with modified content (file replacement) containing a warning message. But with protocol-only notifications, a protocol-specific error code may be returned to the client. This way, the client determines that a virus was detected rather than interpreting that a file transfer succeeded.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Configuring Protocol-Only Virus-Detected Notifications (CLI Procedure)

To configure protocol-only virus-detected notifications, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine profile name {
  notification-options {
    virus-detection {
      type { protocol-only | message }
    }
    fallback-block {
      type { protocol-only | message }
    }
  }
}
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

E-Mail Virus-Detected Notifications

- Understanding E-Mail Virus-Detected Notifications on page 628
- Configuring E-Mail Virus-Detected Notifications (CLI Procedure) on page 628

Understanding E-Mail Virus-Detected Notifications

For mail protocols (SMTP, POP3, IMAP), e-mail notification is used to notify the sender or the recipient about the detected viruses or the scanning errors. There are three settings for e-mail notifications:

- virus-detection/notify-mail-sender — This setting is used when a virus is detected. If it is enabled, an e-mail is sent to the sender upon virus detection.
- fallback-block/notify-mail-sender — This setting is used when other scan codes or scanning errors are returned and the message is dropped. If it is enabled, an e-mail is sent to the sender when an error code is returned.
- fallback-non-block/notify-mail-recipient — This setting is used when other scan codes or scanning errors are returned and the message is passed. If it is enabled, the e-mail sent to the recipient is tagged when an error code is returned.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Configuring E-Mail Virus-Detected Notifications (CLI Procedure)

To configure the system to send e-mail notifications when viruses are detected, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine profile name {  
  notification-options {  
    virus-detection {  
      notify-mail-sender  
    }  
    fallback-block {  
      notify-mail-sender  
    }  
    fallback-non-block {  
      notify-mail-recipient  
    }  
  }  
}
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Custom Message Virus-Detected Notifications

- Understanding Custom Message Virus-Detected Notifications on page 628
- Configuring Custom Message Virus-Detected Notifications (CLI Procedure) on page 629

Understanding Custom Message Virus-Detected Notifications

Custom message notifications are mainly used in file replacement or in a response message when the antivirus scan result is to drop the file. When using custom messages, you can provide a customized message in the message content you can define customized subject tags.



NOTE: Custom-message in fallback-nonblock is used only by mail protocols.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Configuring Custom Message Virus-Detected Notifications (CLI Procedure)

To configure the system to send custom messages when viruses are detected, use the following CLI configuration statements:

```
security utm feature-profile anti-virus kaspersky-lab-engine profile name {
  notification-options {
    virus-detection {
      custom-message msg
      custom-message-subject subject-msg
    }
    fallback-block {
      custom-message msg
      custom-message-subject subject-msg
    }
    fallback-non-block {
      custom-message msg
      custom-message-subject subject-msg
    }
  }
}
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Scanning Fallback Options

- Understanding Antivirus Scanning Fallback Options on page 629
- Example: Configuring Antivirus Scanning Fallback Options (CLI) on page 630

Understanding Antivirus Scanning Fallback Options

Fallback options tell the system how to handle the errors returned by either the scan engine or the scan manager. The following is a list of possible errors and the default fallback actions for those error types:

- Scan engine is not ready (engine-not-ready)

The scan engine is initializing itself, for example, loading the signature database. During this phase, it is not ready to scan a file. A file could either pass or be blocked according to this setting. The default action is BLOCK.

- Corrupt file (corrupt-file)

Corrupt file is the error returned by the scan engine when engine detects a corrupted file. The default action is PASS.

- Decompression layer (decompress-layer)

Decompress layer error is the error returned by the scan engine when the scanned file has too many compression layers. The default action is BLOCK.

- Password protected file (password-file)

Password protected file is the error returned by the scan engine when the scanned file is protected by a password. The default action is PASS.

- Max content size (content-size)

If the content size exceeds a set limit, the content is passed or blocked depending on the max-content-size fallback option. The default action is BLOCK.

- Too many requests (too-many-requests)

If the total number of messages received concurrently exceeds the device limits, the content is passed or blocked depending on the too-many-request fallback option. The default action is BLOCK. (The allowed request limit is not configurable.)

- Timeout

Scanning a complex file could consume resources and time. If the time it is taking to scan exceeds the timeout setting in the antivirus profile, the processing is aborted and the content is passed or blocked without completing the virus checking. The decision is made based on the timeout fallback option. The default action is BLOCK.

- Out of resources (out-of-resources)

Virus scanning requires a great deal of memory and CPU resources. Due to resource constraints, memory allocation requests can be denied by the system. This failure could be returned by either scan engine (as a scan-code) or scan manager. When out-of-resources occurs, scanning is aborted. The default action is BLOCK.

- Default

All the errors other than those in the above list fall into this category. This could include either unhandled system exceptions (internal errors) or other unknown errors. The default action is BLOCK.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Configuring Antivirus Scanning Fallback Options (CLI)

To configure antivirus fallback scanning options, enter the following CLI configuration statements. The following example configures fallback scanning options for the kasprof1 profile:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
  fallback-options content-size block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
  fallback-options corrupt-file block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
  fallback-options decompress-layer block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
  fallback-options default block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
  fallback-options engine-not-ready block
```

```

user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options out-of-resources block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options password-file block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine
kasprof1fallback-options timeout block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options too-many-requests block

```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Full Antivirus Configuration Overview

When configuring antivirus protection, you must first create the antivirus custom objects you are using. Those custom objects may include the MIME pattern list, MIME exception list, and the filename extension list. Once you have created your custom objects, you can configure full antivirus protection, including intelligent prescreening, and content size limits.

To configure full file-based antivirus protection:

1. Configure UTM custom objects for the UTM feature. The following example enables the mime-pattern, filename-extension, url-pattern, and custom-url-category custom-objects:

```

user@host# set security utm custom-objects mime-pattern
user@host# set security utm custom-objects filename-extension
user@host# set security utm custom-objects url-pattern
user@host# set security utm custom-objects custom-url-category

```

2. Configure the main feature parameters using feature profiles. The following example enables options using the anti-virus feature profile:

```

user@host# set security utm feature-profile anti-virus kaspersky-lab-engine
pattern-update
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
fallback-options
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
notification-options
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
scan-options
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
trickling
user@host# set security utm feature-profile anti-virus mime-whitelist
user@host# set security utm feature-profile anti-virus url-whitelist

```

3. Configure a UTM policy for each protocol and attach this policy to a profile. The following example configure the utmp2 UTM policy for the HTTP protocol:

```

user@host# set security utm utm-policy utmp2 anti-virus http-profile http1

```

4. Attach the UTM policy to a security policy. The following example attaches the utmp2 UTM policy to the p2 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p2 then
permit application-services utm-policy utmp2
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Configuring Full Antivirus (J-Web Procedure)

- Configuring Full Antivirus Custom Objects (J-Web Procedure) on page 632
- Configuring Full Antivirus Feature Profiles (J-Web Procedure) on page 634
- Configuring Full Antivirus UTM Policies (J-Web Procedure) on page 637
- Attaching Full Antivirus UTM Policies to Security Policies (J-Web Procedure) on page 638

Configuring Full Antivirus Custom Objects (J-Web Procedure)

To configure antivirus protection, you must first create your custom objects (MIME Pattern List, Filename Extension List, URL Pattern List, and Custom URL Category List).

Configure a MIME pattern list custom object:

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the MIME Pattern List tab, click the **Add** button to create MIME pattern lists.
3. In the Add MIME Pattern pop-up window, next to **MIME Pattern Name**, enter a unique name.



NOTE: Keep in mind that you are creating a MIME whitelist and a MIME exception list (if necessary). Both MIME lists appear in the MIME Whitelist and Exception MIME Whitelist fields when you configure antivirus. Therefore, the MIME list names you create should be as descriptive as possible.

4. Next to **MIME Pattern Value**, enter the MIME pattern.
5. Click **Add** to add your MIME pattern to the Values list box. Within this box, you can also select an entry and use the Delete button to delete it from the list. Continue to add MIME patterns in this manner.
6. Optionally, create a new MIME list to act as an exception list. The exception list is generally a subset of the main MIME list.
7. Click **OK** to check your configuration and save the selected values as part of the MIME list, then click **Commit Options>Commit**.
8. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configure a filename extension list custom object:

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the Filename Extension List tab, click the **Add** button to create filename extension lists.
3. Next to **File Extension Name**, enter a unique name. This name appears in the Scan Option By Extension list when you configure an antivirus profile.
4. In the **Available Values** box, select one or more default values (press Shift to select multiple concurrent items or press Ctrl to select multiple separate items) and click the right arrow button to move the value or values to the Selected Values box.
5. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
6. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configure a URL pattern list custom object:



NOTE: Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure a custom URL category list.

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the URL Pattern List tab, click the **Add** button to create URL pattern lists.
3. Next to URL Pattern Name, enter a unique name. This name appears in the Custom URL Category List Custom Object page for selection.
4. Next to **URL Pattern Value**, enter the URL or IP address you want added to the list for bypassing scanning.

When entering the URL pattern, note the following wildcard character support:

- The `*\.[]\?` wildcard characters are supported.
- You must precede all wildcard URLs with **http://**.
- You can only use the asterisk `*` wildcard character if it is at the beginning of the URL and is followed by a period.
- You can only use the question mark `?` wildcard character at the end of the URL.
- The following wildcard syntax IS supported: **http://*.juniper.net**, **http://www.juniper.ne?**, **http://www.juniper.n??**.
- The following wildcard syntax is NOT supported: ***juniper.net** , **www.juniper.ne?**, **http://*juniper.net**, **http://***.

5. Click **Add** to add your URL pattern to the Values list box. The list can contain up to 8192 items. You can also select an entry and use the Delete button to delete it from the list. Continue to add URLs or IP addresses in this manner.
6. Click **OK** to check your configuration and save the selected values as part of the URL pattern list you have created, then click **Commit Options>Commit**.
7. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configure a custom URL category list custom object:



NOTE: Because you use URL Pattern Lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure a custom URL category list.

1. Select **Configure>Security>UTM>Custom Objects**.
2. In the URL Category List tab, click **Add** to create URL category lists.
3. Next to **URL Category Name**, enter a unique name. This name appears in the URL Whitelist list when you configure antivirus global options.
4. In the **Available Values** box, select a URL Pattern List name from the list for bypassing scanning and click the right arrow button to move it to the Selected Values box.
5. Click **OK** to check your configuration and save the selected values as part of the URL list that you have created, then click **Commit Options>Commit**.

Click **OK** to save the selected values as part of the custom URL list you have created.
6. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configuring Full Antivirus Feature Profiles (J-Web Procedure)

After you have created your custom object, configure an antivirus feature profile:

1. Select **Configure>Security>UTM>Global options**.
2. In the Anti-Virus tab, next to **MIME whitelist**, select the custom object you created from the list.
3. Next to **Exception MIME whitelist**, select the custom object you created from the list.
4. Next to **URL Whitelist**, select the custom object you created from the list.
5. In the **Engine Type** section, select the type of engine you are using. For full antivirus protection, you should select **Kaspersky Lab**.
6. In the Kaspersky Lab Engine Option section, in the **Pattern update URL** box, enter the URL for the pattern database.



NOTE: The URL is <http://update.juniper-updates.net/AV/<device version>> and you should not change it.

7. Next to **Pattern update interval**, enter the time interval, in seconds, for automatically updating the pattern database in the box. The default interval is 60.
8. Select whether you want the pattern file to update automatically (**Auto update**) or not (**No Auto update**).
9. Click **OK** to save the selected values.
10. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in a pop-up window that appears to discover why.
11. Under Security, in the left pane, select **Anti-Virus**.
12. In the right window, click **Add** to create a profile for the antivirus Kaspersky Lab Engine. (To edit an existing item, select it and click the **Edit** button.)
13. Next to **Profile name**, enter a unique name for this antivirus profile.
14. Select the **Profile Type**. In this case, select **Kaspersky**.
15. Next to **Trickling timeout**, enter timeout parameters.



NOTE: Trickling applies only to HTTP. HTTP trickling is a mechanism used to prevent the HTTP client or server from timing out during a file transfer or during antivirus scanning.

16. Next to Intelligent prescreening, select **Yes** or **No**.



NOTE: Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for mail protocols (SMTP, POP3, IMAP, and HTTP POST).

17. In the Scan Options section, next to Intelligent prescreening, select **Yes** if you are using it.



NOTE: Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for mail protocols (SMTP, POP3, IMAP, and HTTP POST).

18. Next to **Content Size Limit**, enter content size parameters. The content size check occurs before the scan request is sent. The content size refers to accumulated TCP payload size.
19. Next to **Scan engine timeout**, enter scanning timeout parameters.
20. Next to **Decompress Layer Limit**, enter decompression layer limit parameters.
21. In the Scan mode section, select either **Scan all files**, if you are scanning all content, or **Scan files with specified extension**, if you are scanning by file extensions.



NOTE: If you select Scan files with specified extension, you must select a filename extension list custom object from the Scan engine filename extension list that appears.

22. Select the **Fallback settings** tab.
23. Next to Default (fallback option), select **Log and permit** or **Block** from the list. In most cases, Block is the default fallback option.
24. Next to Corrupt File (fallback option), select **Log and permit** or **Block** from the list.
25. Next to Password File (fallback option), select **Log and permit** or **Block** from the list.
26. Next to Decompress Layer (fallback option), select **Log and permit** or **Block** from the list.
27. Next to Content Size (fallback option), select **Log and permit** or **Block** from the list.
28. Next to Engine Not Ready (fallback option), select **Log and permit** or **Block** from the list.
29. Next to Timeout (fallback option), select **Log and permit** or **Block** from the list.
30. Next to Out Of Resources (fallback option), select **Log and permit** or **Block** from the list.
31. Next to Too Many Request (fallback option), select **Log and permit** or **Block** from the list.
32. Select the **Notification options** tab.
33. In the Fallback block section, next to Notification type, select **Protocol Only** or **Message** to select the type of notification that is sent when a fallback option of block is triggered.
34. Next to Notify mail sender, select **Yes** or **No**.
35. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).
36. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message).
37. In the Fallback non block section, next to Notify mail recipient, select **Yes** or **No**.
38. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).
39. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message).
40. Select the **Notification options cont** tab.
41. In the Virus detection section, next to Notification type, select **Protocol Only** or **Message** to select the type of notification that is sent when a fallback option of block is triggered.
42. Next to Notify mail sender, select **Yes** or **No**.

43. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).
44. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message). The limit is 255 characters.
45. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
46. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.



NOTE: You create a separate antivirus profile for each antivirus protocol. These profiles may basically contain the same configuration information, but when you are creating your UTM policy for an antivirus profile, the UTM policy configuration page provides separate antivirus profile selection fields for each supported protocol.

Configuring Full Antivirus UTM Policies (J-Web Procedure)

After you have created an antivirus feature profile, configure a UTM policy to which you can attach the feature profile:

1. Select **Configure>Security>Policy>UTM Policies**.
2. From the UTM policy configuration window, click **Add** to configure a UTM policy. This action takes you to the policy configuration pop-up window.
3. Select the **Main** tab in pop-up window.
4. In the **Policy name** box, enter a unique name for the UTM policy.
5. In the **Session per client limit** box, enter a session per client limit from 0 to 20000 for this UTM policy.
6. In the **Session per client over limit** list, select the action that the device should take when the session per client limit for this UTM policy is exceeded. Options include **Log and permit** and **Block**.
7. Select the **Anti-Virus profiles** tab in the pop-up window.
8. Select the appropriate profile you have configured from the list for the corresponding protocol listed.
9. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
10. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Attaching Full Antivirus UTM Policies to Security Policies (J-Web Procedure)

After you create a UTM policy, create a security policy and attach the UTM policy to the security policy:

1. Select **Configure>Security>Policy>FW Policies**.
2. From the Security Policy window, click **Add** to configure a security policy with UTM. This action takes you to the policy configuration pop-up window.
3. In the Policy tab, enter a name in the **Policy Name** box.
4. Next to **From Zone**, select a zone from the list.
5. Next to **To Zone**, select a zone from the list.
6. Choose a **Source Address**.
7. Choose a **Destination Address**.
8. Choose an application by selecting **junos-protocol** (for all protocols that support antivirus scanning) in the Application Sets box and clicking the —> button to move it to the Matched box.
9. Next to Policy Action, select **Permit**.



NOTE: When you select Permit for Policy Action, several additional fields become available in the Applications Services tab, including UTM Policy.

10. Select the **Application Services** tab in the pop-up window.
11. Next to **UTM Policy**, select the appropriate policy from the list. This action attaches your UTM policy to the security policy.
12. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
13. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

You must activate your new policy to apply it.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Configuring Full Antivirus (CLI)

- Example: Configuring Full Antivirus Custom Objects (CLI) on page 639
- Example: Configuring Full Antivirus Feature Profiles (CLI) on page 640
- Example: Configuring Full Antivirus UTM Policies (CLI) on page 643
- Example: Attaching Full Antivirus UTM Policies to Security Policies (CLI) on page 643

Example: Configuring Full Antivirus Custom Objects (CLI)

To configure antivirus protection, you must first create your custom objects (MIME Pattern List, Filename Extension List, URL Pattern List, and Custom URL Category List).

1. Configure the filename-extension custom object by first creating a name for the list. The following example creates the extlist1 custom object:

```
user@host# set security utm custom-objects filename-extension extlist1
```



NOTE: The Kaspersky scan engine ships with a read-only default extension list that you can use.

2. Add extensions to the list. The following example adds the zip, js, and vbs extensions to the extlist1 custom object:

```
user@host# set security utm custom-objects filename-extension extlist1 value [zip
js vbs]
```

3. Create MIME lists. The following example creates the avmime1 and ex-avmime1 lists:

```
user@host# set security utm custom-objects mime-pattern avmime1
user@host# set security utm custom-objects mime-pattern ex-avmime1
```

4. Add MIME patterns to the lists. The following example adds patterns to the avmime1 and ex-avmime1 lists:

```
user@host# set security utm custom-objects mime-pattern avmime1 value
[video/quicktime image/x-portable-anymap x-world/x-vrml]
user@host# set security utm custom-objects mime-pattern ex-avmime1 value
[video/quicktime-inappropriate]
```



NOTE: Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure custom URL category lists.

5. Configure a URL pattern list custom object specifying a list of URLs that you want the device to bypass during scanning. The following example creates the urlist1 list:

```
user@host# set security utm custom-objects url-pattern urlist1 value
[http://www.url.com 5.6.7.8]
```

When entering the URL pattern, note the following wildcard character support:

- The `*\.[]\?` wildcard characters are supported.
- You must precede all wildcard URLs with `http://`.
- You can only use the asterisk `*` wildcard character if it is at the beginning of the URL and is followed by a period.
- You can only use the question mark `?` wildcard character at the end of the URL.

- The following wildcard syntax IS supported: `http://*.juniper.net`, `http://www.juniper.net?`, `http://www.juniper.n??`.
 - The following wildcard syntax is NOT supported: `*juniper.net`, `www.juniper.net?`, `http://*juniper.net`, `http://*`.
6. Configure a custom URL category list custom object using the URL pattern list you created. The following example adds the `urllist1` list to the `custurl1` custom object:
- ```
user@host# set security utm custom-objects custom-url-category custurl1 value
urllist1
```

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Example: Configuring Full Antivirus Feature Profiles (CLI)

After you have created your custom object, configure an antivirus feature profile:

1. Select and configure the engine type. Because you are configuring “full antivirus,” you select the Kaspersky-Lab-Engine and then designate the pattern update interval. The default full file-based antivirus pattern-update interval is 60 minutes. You can choose to leave this default as is or you can change it. You can also force a manual update, if necessary. The following example sets the engine type to Kaspersky-Lab-Engine and sets the update interval to 20:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine
pattern-update interval 20
```



**NOTE:** The command for changing the URL for the pattern database is:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine
pattern-update url http://.
```

The default URL is `http://update.juniper-update.net/AV/<device version>`. You should not change this URL unless you are experiencing problems with it and have called for support.

- 
2. Configure the device to notify a specified administrator when patterns are updated. The following example enables an e-mail notification with a custom message and a custom subject line:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine
pattern-update email-notify admin-email administrator@juniper.net
custom-message "pattern file was updated" custom-message-subject "AV pattern
file updated"
```

3. Configure a profile for the Kaspersky Lab engine. The following example creates the `kasprof1` profile:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine profile
kasprof1
```



4. Configure a list of fallback options as block or log-and-permit. In most cases, the default is to block. You can use the default settings or you can change them. The following example configures fallback options as block for the `kasprof1` profile:

```

user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options content-size block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options corrupt-file block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options decompress-layer block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options default block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options engine-not-ready block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options out-of-resources block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options password-file block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options timeout block
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
fallback-options too-many-requests block

```

5. Configure the notification options. You can configure notifications for both fallback blocking and fallback nonblocking actions and for virus detection. You configure a custom message for the fallback blocking action and send a notification. The following example configures the device to send the `***virus-found***` notification for blocked traffic:

```

user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
notification-options fallback-block custom-message ***virus-found***
notify-mail-sender

```

6. Configure a custom subject line for the custom message notification for both the sender and the recipient. The following example configures the device to add "Antivirus Alert" to the message subject line:

```

user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
notification-options fallback-block custom-message-subject "Antivirus Alert"
notify-mail-sender

```

7. Configure a notification for protocol-only virus detection and send a notification. The following example configures the protocol-only virus detection for the `kasprof1` profile:

```

user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
notification-options virus-detection type protocol-only notify-mail-sender

```

8. Configure scan options. The following example configures the device to perform a TCP payload content size check before the scan request is sent:

```

user@host# set security utm feature-profile anti-virus kaspersky-lab-engine
kasprof1scan-options content-size-limit 20000

```

9. Configure the decompression layer limit. The following example configures the device to decompress 3 layers of nested compressed files before it executes the virus scan:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
scan-options decompress-layer-limit 3
```

10. Configure intelligent prescreening. It is either on or off. The following example enables intelligent prescreening for the kasprof1 profile:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
scan-options intelligent-prescreening
```



**NOTE:** Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for mail protocols (SMTP, POP3, IMAP) and HTTP POST.

The following example disables intelligent prescreening for the kasprof1 profile:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
scan-options no-intelligent-prescreening
```

11. Configure scan extension settings. You can select the default list (junos-default-extension) or you can select an extension list you created as a custom object. The following example enables the extlist1 for the kasprof1 profile:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
scan-options scan-extension extlist1
```

12. Configure the scan mode setting. You can choose to scan all files or only files with the extensions that you specify. The following example uses the scan by-extension option to configure the device to use a custom extension list:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
scan-options scan-mode by-extension
```

13. Configure the timeout settings. The following example sets the scan-mode timeout to 1800 seconds:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
scan-options scan-mode timeout 1800
```

14. Configure trickling settings. The following example indicates that if the device receives a packet within a 600 second period during a file transfer or while performing an antivirus scan, it should not timeout:

```
user@host# set security utm feature-profile anti-virus kaspersky-lab-engine kasprof1
trickling timeout 600
```



**NOTE:** Trickling applies only to HTTP.

15. Configure the antivirus scanner to use MIME bypass lists and exception lists. You can use your own custom object lists, or you can use the default list that ships with the device called junos-default-bypass-mime. The following examples enable the avmime1 and ex-avmime1 lists:

```
user@host# set security utm feature-profile anti-virus mime-whitelist list avmime1
user@host# set security utm feature-profile anti-virus mime-whitelist list avmime1
exception ex-avmime1
```

16. Configure the antivirus module to use URL bypass lists. If you are using a URL whitelist, this is a custom URL category you have previously configured as custom object. The following example enables the custurl1 bypass list:

```
user@host# set security utm feature-profile anti-virus url-whitelist custurl1
```



**NOTE:** URL whitelists are valid only for HTTP traffic.

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Configuring Full Antivirus UTM Policies (CLI)

After you have created an antivirus feature profile, configure a UTM policy to which you can attach the feature profile:

1. Create a UTM policy for HTTP antivirus scanning. The following example creates the utmp2 policy:

```
user@host# set security utm utm-policy utmp2
```

2. Attach the policy to profile. The following example attaches the utmp2 policy to the kasprofile1 HTTP profile:

```
user@host# set security utm utm-policy utmp2 anti-virus http-profile kasprofile1
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Attaching Full Antivirus UTM Policies to Security Policies (CLI)

After you create a UTM policy, create a security policy and attach the UTM policy to the security policy:

1. Enable and configure the security policy. The following example enables the p2 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p2 match
source-address any
```

```
user@host# set security policies from-zone trust to-zone untrust policy p2 match
destination-address any
```

```
user@host# set security policies from-zone trust to-zone untrust policy p2 match
application junos-http
```

2. Attach the UTM policy to the security policy. The following example attaches the utmp2 UTM policy to the p2 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p2 then
permit application-services utm-policy utmp2
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Monitoring Antivirus Sessions and Scan Results

---

The antivirus module provides functions which allow you to use the CLI to check the system settings and the status of scan engine. It also provides functions to check the ongoing antivirus sessions and antivirus statistics.

- Monitoring Antivirus Scan Engine Status on page 644
- Monitoring Antivirus Session Status on page 644
- Monitoring Antivirus Scan Results on page 645

### Monitoring Antivirus Scan Engine Status

**Purpose** Using the CLI, you can view the following scan engine status items:

Antivirus license key status

- View license expiration dates.

Scan engine status and settings

- View last action result.
- View default file extension list.

Antivirus pattern update server settings

- View update URL (HTTP or HTTPS-based).
- View update interval.

Antivirus pattern database status

- View auto update status.
- View last result of database loading.
- If the download completes, view database version timestamp virus record number.
- If the download fails, view failure reason.

**Action** In the CLI, enter the `user@host> show security utm anti-virus status` command.

Example status result:

```
AV Key Expire Date: 03/01/2010 00:00:00
Update Server: http://update.juniper-updates.net/AV/SRX210
interval: 60 minutes
auto update status: next update in 12 minutes
last result: new database loaded
AV signature version: 12/21/2008 00:35 GMT, virus records: 154018
Scan Engine Info: last action result: No error(0x00000000)
```

### Monitoring Antivirus Session Status

**Purpose** Using the CLI, you can view the following session status items:

Antivirus session status displays a snapshot of current antivirus sessions. It includes

- Maximum supported antivirus session numbers.
- Total allocated antivirus session numbers.
- Total freed antivirus session numbers.
- Current active antivirus session numbers.

**Action** In the CLI, enter the `user@host> show security utm session status` command.

## Monitoring Antivirus Scan Results

**Purpose** View statistics for antivirus requests, scan results, and fallback counters.

Scan requests provide

- The total number of scan request forwarded to the engine.
- The number of scan request being pre-windowed.
- The number of scan requests using scan-all mode.
- The number of scan requests using scan-by-extension mode.

Scan code counters provide

- Number of clean files.
- Number of infected files.
- Number of password protected files.
- Number of decompress layers.
- Number of corrupt files.
- When the engine is out of resources.
- When there is an internal error.

Fallback applied status provides either a log-and-permit or block result when the following has occurred

- Scan engine not ready.
- Maximum content size reached.
- Too many requests.
- Password protected file found.
- Decompress layer too large.
- Corrupt file found.
- Timeout occurred.

- Out of resources.
- Other.

**Action** To view antivirus scan results using the CLI editor, enter the **user@host> show security utm anti-virus statistics status** command.

To view antivirus scan results using J-Web:

1. Select **Monitor>UTM>Anti-Virus**.

The following information becomes viewable in the right pane.

Antivirus license key status

- View license expiration dates.

Antivirus pattern update server settings

- View update URL (HTTP or HTTPS-based).
- View update interval.

Antivirus pattern database status

- View auto update status.
- View last result of database loading.
- If the download completes, view database version timestamp virus record number.
- If the download fails, view failure reason.

Antivirus statistics provide

- The number of scan request being pre-windowed.
- The total number of scan request forwarded to the engine.
- The number of scan requests using scan-all mode.
- The number of scan requests using scan-by-extension mode.

Scan code counters provide

- Number of clean files.
- Number of infected files.
- Number of password protected files.
- Number of decompress layers.
- Number of corrupt files.
- When the engine is out of resources.
- When there is an internal error.

Fallback applied status provides either a log-and-permit or block result when the following has occurred

- Scan engine not ready.
  - Password protected file found.
  - Decompress layer too large.
  - Corrupt file found.
  - Out of resources.
  - Timeout occurred.
  - Maximum content size reached.
  - Too many requests.
  - Other.
2. You can click the **Clear Anti-Virus Statistics** button to clear all current viewable statistics and begin collecting new statistics.

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*





# Express Antivirus Protection

- Express Antivirus Protection Overview on page 649
- Express Antivirus Scanner Pattern Database on page 651
- Express Antivirus Configuration Overview on page 653
- Configuring Express Antivirus (J-Web Procedure) on page 653
- Example: Configuring Express Antivirus (CLI) on page 659

## Express Antivirus Protection Overview

---

Express antivirus scanning is offered as a less CPU intensive alternative to the full file-based antivirus feature. Express antivirus supports the same protocols as full antivirus and functions in much the same manner, however, it has a smaller memory footprint, compatible with the smaller system memory present on lower end devices.



**NOTE:** If you switch from express antivirus protection to full file-based antivirus protection, you must reboot the device in order for full file-based antivirus to begin working.

This topic includes the following sections:

- Express Antivirus Packet-Based Scanning Versus File-Based Scanning on page 649
- Express Antivirus Expanded MIME Decoding Support on page 650
- Express Antivirus Scan Result Handling on page 650
- Express Antivirus Intelligent Prescreening on page 650
- Express Antivirus Limitations on page 650

## Express Antivirus Packet-Based Scanning Versus File-Based Scanning

Express antivirus uses a different antivirus scan engine than the full file-based antivirus feature and a different back-end hardware engine to accelerate pattern matching for higher data throughput.

The packet based scanning done by express antivirus provides virus scanning data buffers without waiting for entire file to be received by the firewall, whereas the file-based scanning done by full antivirus can only start virus scanning when entire file is received.

## Express Antivirus Expanded MIME Decoding Support

Express antivirus offers MIME decoding support for HTTP, POP3, SMTP, and IMAP. MIME decoding support includes the following for each supported protocol:

- Multi-part and nested header decoding
- Base64 decoding, printed quote decoding, and encoded word decoding (in the subject field)

## Express Antivirus Scan Result Handling

With express antivirus, the TCP traffic is closed gracefully when a virus is found and the data content is dropped.



**NOTE:** Express antivirus supports the following fail mode options: default, engine-not-ready, out-of-resource, and too-many-requests. Fail mode handling of supported options with express antivirus is much the same as with full antivirus.

---

## Express Antivirus Intelligent Prescreening

Intelligent prescreening functionality is identical in both express antivirus and full antivirus.

## Express Antivirus Limitations

Express antivirus has the following limitations when compared to full antivirus functionality:

- Express antivirus provides limited support for the scanning of file archives and compressed file formats. Express antivirus can only support gzip, deflate and compressed compressing formats.
- Express antivirus provides limited support for decompression. Decompression is only supported with HTTP (supports only gzip, deflate, and compress for HTTP and only supports one layer of compression) and POP3 (supports only gzip for POP3 and only supports one layer of compression).
- Express antivirus does not support scanning by extension.
- Express antivirus scanning is interrupted when the scanning database is loading.
- Express antivirus may truncate a warning message if a virus has been detected and the replacement warning message that is sent is longer than the original content it is replacing.



NOTE: Because express antivirus does only packet based string matching, if you use the standard EICAR file to test express antivirus, you will see false positives. To avoid these false positives, Juniper has disabled scanning on the standard EICAR file to create a modified EICAR file for testing express antivirus. You can download this modified EICAR file from the following links:

<http://www.juniper.net/security/avtest/ss-eicar.txt>

<http://www.juniper.net/security/avtest/ss-eicar.com>

<http://www.juniper.net/security/avtest/ss-eicar.zip>

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Express Antivirus Scanner Pattern Database

- Understanding Express Antivirus Scanner Pattern Updates on page 651
- Example: Automatically Updating Express Antivirus Patterns (J-Web) on page 652
- Example: Automatically Updating Express Antivirus Patterns (CLI) on page 652
- Manually Updating, Reloading, and Deleting Express Antivirus Patterns (CLI Procedure) on page 652

### Understanding Express Antivirus Scanner Pattern Updates

Express antivirus uses a different signature database than the full antivirus signature database. The express antivirus signature database is called Juniper Express antivirus database and it is compatible with the hardware engine. The express signature database targets only critical viruses and malware, including worms, Trojans, and spyware. This is a smaller sized database, providing less coverage than the full antivirus signature database.

The express antivirus pattern database is updated over HTTP or HTTPS and can occur automatically or manually. This is similar functionality to that found in full antivirus with some minor differences:

- With express antivirus, the signature database auto-update interval, is once a day.
- With express antivirus, there is no support for the downloading of multiple database types.
- With express antivirus, during database loading, all scan operations are interrupted. Scan operations for existing traffic flows are stopped and no new scan operations are initiated for newly established traffic flows. You can specify the desired action for this interruption period using the **fall-back** parameter for **engine-busy-loading-database**. The available actions are **block** or **log-and-permit**.
- By default, the URL for express antivirus is <http://update.juniper-updates.net/EAV/SRX210>. "SRX210" in the URL is the platform name. This part of the URL is different and platform specific for each platform. (Other

than the platform name, you should not change this URL unless you are experiencing problems with it and have called for support.)



**NOTE:** Once your subscription expires, you have a 30 day grace period during which you can continue to update the antivirus pattern file. Once that grace period expires, the update server no longer permits antivirus pattern file updates.

The express Antivirus scanning feature is a separately licensed subscription service. When your antivirus license key expires, you can continue to use locally stored antivirus signatures. But in that case, if the local database is deleted, antivirus scanning is disabled.

---

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Automatically Updating Express Antivirus Patterns (J-Web)

In this example, you configure the security device to update the pattern file automatically every 120 minutes. (The default antivirus pattern-update interval is once a day.)

To automatically update antivirus patterns:

1. Select **Configure>Security>UTM>Anti-Virus**.
2. Next to Interval, in the Juniper Express Engine section, enter **120** in the box.
3. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Automatically Updating Express Antivirus Patterns (CLI)

In this example, you configure the security device to update the pattern file automatically every 120 minutes. (The default antivirus pattern-update interval is once a day.)

To automatically update antivirus patterns, enter the following CLI statement:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
pattern-update interval 120
```

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Manually Updating, Reloading, and Deleting Express Antivirus Patterns (CLI Procedure)

To manually update antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus juniper-express-engine pattern-update
```

To manually reload antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus juniper-express-engine pattern-reload
```

To manually delete antivirus patterns, enter the following CLI statement:

```
user@host> request security utm anti-virus juniper-express-engine pattern-delete
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Express Antivirus Configuration Overview

For each UTM feature, you should configure feature parameters in the following order:

1. Configure UTM custom objects for the UTM features. The following example enables the mime-pattern, url-pattern, and custom-url-category custom objects:

```
user@host# set security utm custom-objects mime-pattern
user@host# set security utm custom-objects url-pattern
user@host# set security utm custom-objects custom-url-category
```

2. Configure main feature parameters using feature profiles. The following examples enables the anti-virus feature profile:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
```

3. Configure a UTM policy for each protocol and attach this policy to a profile. The following example creates the utmp3 UTM policy for the HTTP protocol:

```
user@host# set security utm utm-policy utmp3 anti-virus http-profile http1
```

4. Attach the UTM policy to a security policy. The following example attaches the utmp3 UTM policy to the p3 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p3 then
permit application-services utm-policy utmp3
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Configuring Express Antivirus (J-Web Procedure)

- Configuring Express Antivirus Custom Objects (J-Web Procedure) on page 653
- Configuring Express Antivirus Feature Profiles (J-Web Procedure) on page 655
- Configuring Express Antivirus UTM Policies (J-Web Procedure) on page 658
- Attaching Express Antivirus UTM Policies to Security Policies (J-Web Procedure) on page 658

### Configuring Express Antivirus Custom Objects (J-Web Procedure)

To configure express antivirus protection using the J-Web configuration editor, you must first create your custom objects (MIME pattern list, URL pattern list, and custom URL category list).

Configure a MIME pattern list custom object as follows:

1. Select **Configure>Security>UTM Custom Objects**.
2. From the MIME Pattern List tab, click **Add** to create MIME pattern lists.

3. In the Add MIME Pattern pop-up window, next to **MIME Pattern Name**, enter a unique name.



**NOTE:** Keep in mind that you are creating a MIME whitelist and a MIME exception list (if necessary). Both MIME lists appear in the MIME Whitelist and Exception MIME Whitelist fields when you configure antivirus. Therefore, the MIME list names you create should be as descriptive as possible.

4. Next to **MIME Pattern Value**, enter the MIME pattern.
5. Click **Add** to add your MIME pattern to the Values list box. Within this box, you can also select an entry and use the Delete button to delete it from the list. Continue to add MIME patterns in this manner.
6. Optionally, create a new MIME list to act as an exception list. The exception list is generally a subset of the main MIME list.
7. Click **OK** to check your configuration and save the selected values as part of the MIME list, then click **Commit Options>Commit**.
8. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configure a URL pattern list custom object as follows:



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure a custom URL category list.

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the URL Pattern List tab, click **Add** to create URL pattern lists.
3. Next to **URL Pattern Name**, enter a unique name. This name appears in the Custom URL Category List Custom Object page for selection.
4. Next to **URL Pattern Value**, enter the URL or IP address you want added to list for bypassing scanning.

When entering the URL pattern, note the following wildcard character support:

- The `\*\.[\]\?*` wildcard characters are supported.
- You must precede all wildcard URLs with `http://`.
- You can only use the asterisk `*` wildcard character if it is at the beginning of the URL and is followed by a period.
- You can only use the question mark `?` wildcard character at the end of the URL.

- The following wildcard syntax IS supported: **http://\*.juniper.net**, **http://www.juniper.ne?**, **http://www.juniper.n??**.
  - The following wildcard syntax is NOT supported: **\*juniper.net** , **www.juniper.ne?**, **http://\*juniper.net**, **http://\***.
5. Click **Add** to add your URL pattern to the Values list box. The list can contain up to 8192 items. You can also select an entry and use the Delete button to delete it from the list. Continue to add URLs or IP addresses in this manner.
  6. Click **OK** to check your configuration and save the selected values as part of the URL pattern list, then click **Commit Options>Commit**.
  7. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configure a custom URL category list custom object using the URL pattern list that you created:

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the URL Category List tab, click **Add** to create URL category lists.
3. Next to **URL Category Name**, enter a unique name. This name appears in the URL Whitelist list when you configure antivirus global options.
4. In the Available Values box, select a **URL Pattern List** name from the list for bypassing scanning and click the right arrow button to move it to the Selected Values box.
5. Click **OK** to check your configuration and save the selected values as part of the URL list, then click **Commit Options>Commit**.
6. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

## Configuring Express Antivirus Feature Profiles (J-Web Procedure)

After you create your custom objects, configure the antivirus feature profile:

1. Select **Configure>Security>UTM>Global options**.
2. In the Anti-Virus tab, next to **MIME whitelist**, select the custom object you created from the list.
3. Next to **Exception MIME whitelist**, select the custom object you created from the list.
4. Next to **URL Whitelist**, select the custom object you created from the list.
5. In the Engine Type section, select the type of engine you are using. For express antivirus protection, you should select **Juniper Express**.
6. Next to **Pattern update URL**, enter the URL for the pattern database in the box. Note that the URL is <http://update.juniper-updates.net/EAV/<device version>> and you should not change it.

7. Next to **Pattern update interval**, enter the time interval for automatically updating the pattern database in the box. The default for express antivirus checking is once per day.
8. Select whether you want the pattern file to update automatically (**Auto update**) or not (**No Auto update**).
9. Click **OK** to save the selected values.
10. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.
11. Under Security, in the left pane, select **Anti-Virus**.
12. Click **Add** in the right window to create a profile for the antivirus Juniper Express Engine. To edit an existing item, select it and click **Edit**.
13. In the Main tab, next to **Profile name**, enter a unique name for this antivirus profile.
14. Select the Profile Type. In this case, select **Juniper Express**.
15. Next to **Trickling timeout**, enter timeout parameters.



**NOTE:** Trickling applies only to HTTP. HTTP trickling is a mechanism used to prevent the HTTP client or server from timing out during a file transfer or during antivirus scanning.

- 
16. Next to Intelligent prescreening, select **Yes** or **No**.



**NOTE:** Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for mail protocols (SMTP, POP3, IMAP, and HTTP POST).

- 
17. Next to **Content Size Limit**, enter content size parameters. The content size check occurs before the scan request is sent. The content size refers to accumulated TCP payload size.
  18. Next to **Scan engine timeout**, enter scanning timeout parameters.
  19. Select the **Fallback settings** tab.
  20. Next to Default (fallback option), select **Log and permit** or **Block** from the list. In most cases, Block is the default fallback option.
  21. Next to Decompress Layer (fallback option), select **Log and permit** or **Block** from the list.
  22. Next to Content Size (fallback option), select **Log and permit** or **Block** from the list.
  23. Next to Engine Not Ready (fallback option), select **Log and permit** or **Block** from the list.
  24. Next to Timeout (fallback option), select **Log and permit** or **Block** from the list.
  25. Next to Out of Resource (fallback option), select **Log and permit** or **Block** from the list.



26. Next to Too Many Requests (fallback option), select **Log and permit** or **Block** from the list.
27. Select the **Notification options** tab.
28. In the Fallback block section, next to Notification type, select **Protocol Only** or **Message** to select the type of notification that is sent when a fallback option of block is triggered.
29. Next to Notify mail sender, select **Yes** or **No**.
30. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).
31. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message).
32. In the Fallback non block section, next to Notify mail recipient, select **Yes** or **No**.
33. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).
34. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message).
35. Select the **Notification options cont** tab.
36. In the Virus detection section, next to Notification type, select **Protocol Only** or **Message** to select the type of notification that is sent when a fallback option of block is triggered.
37. Next to Notify mail sender, select **Yes** or **No**.
38. If you selected Yes, next to **Custom Message**, enter text for the message body of your custom message for this notification (if you are using a custom message).
39. Next to **Custom message subject**, enter text to appear in the subject line of your custom message for this notification (if you are using a custom message). The limit is 255 characters.
40. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
41. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up that appears window to discover why.



**NOTE:** You create a separate antivirus profile for each antivirus protocol. These profiles may basically contain the same configuration information, but when you are creating your UTM policy for antivirus, the UTM policy configuration page provides separate antivirus profile selection fields for each supported protocol.

## Configuring Express Antivirus UTM Policies (J-Web Procedure)

After you have created an antivirus feature profile, configure a UTM policy to which you can attach the feature profile:

1. Select **Configure>Security>Policy>UTM Policies**.
2. From the UTM policy configuration window, click **Add** to configure a UTM policy. The policy configuration pop-up window appears.
3. Select the **Main** tab.
4. In the **Policy name** box, enter a unique name.
5. In the **Session per client limit** box, enter a session per client limit from 0 to 20000 for this UTM policy.
6. In the Session per client over limit list, select the action that the device should take when the session per client limit for this UTM policy is exceeded. Options include **Log and permit** and **Block**.
7. Select the **Anti-Virus profiles** tab.
8. Select the appropriate profile you have configured from the list for the corresponding protocol listed.
9. Click **OK**.
10. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

## Attaching Express Antivirus UTM Policies to Security Policies (J-Web Procedure)

After you create a UTM policy, create a security policy and attach the UTM policy to the security policy:

1. Select **Configure>Security>Policy>FW Policies**.
2. From the Security Policy window, click **Add** to configure a security policy with UTM. The policy configuration pop-up window appears.
3. In the Policy tab, enter a name in the **Policy Name** box.
4. Next to Default Policy Action, select one of the following: **Deny-All** or **Permit-All**.
5. Next to **From Zone**, select a zone from the list.
6. Next to **To Zone**, select a zone from the list.
7. Under Zone Direction, click **Add a Policy**.
8. Choose a **Source Address**.
9. Choose a **Destination Address**.
10. Choose an application by selecting **junos-protocol** (for all protocols that support antivirus scanning) in the Application Sets box and clicking the —> button to move it to the Matched box.

11. Next to Policy Action, select **Permit**.



**NOTE:** When you select Permit for Policy Action, several additional fields become available in the Applications Services tab, including UTM Policy.

12. Select the **Application Services** tab.
13. Next to **UTM Policy**, select the appropriate policy from the list. This action attaches your UTM policy to the security policy.
14. Click **OK**.
15. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options**>**Commit**.
16. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

You must activate your new policy to apply it.

## Example: Configuring Express Antivirus (CLI)

- Example: Configuring Express Antivirus Custom Objects (CLI) on page 659
- Example: Configuring Express Antivirus Feature Profiles (CLI) on page 660
- Example: Configuring Express Antivirus UTM Policies (CLI) on page 663
- Example: Attaching Express Antivirus UTM Policies to Security Policies (CLI) on page 663

## Example: Configuring Express Antivirus Custom Objects (CLI)

To configure antivirus protection using the CLI, you must first create your custom objects.

1. Create MIME lists. The following example creates the avmime2 and ex-avmime2 lists:
 

```
user@host# set security utm custom-objects mime-pattern avmime2
user@host# set security utm custom-objects mime-pattern ex-avmime2
```
2. Add MIME patterns to the lists. The following example adds patterns to the avmime2 and ex-avmime2 lists:
 

```
user@host# set security utm custom-objects mime-pattern avmime2 value
[video/quicktime image/x-portable-anymap x-world/x-vrml]
user@host# set security utm custom-objects mime-pattern ex-avmime2 value
[video/quicktime-inappropriate]
```



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure custom URL category lists.

3. Configure a URL pattern list custom object by creating the list name and adding values to it. The following example creates the urllist2 custom object:

```
user@host# set security utm custom-objects url-pattern urllist2 value
[http://www.juniper.net 1.2.3.4]
```

When entering the URL pattern, note the following wildcard character support:

- The `\*\.[]\?` wildcard characters are supported.
  - You must precede all wildcard URLs with `http://`.
  - You can only use the asterisk `*` wildcard character if it is at the beginning of the URL and is followed by a period.
  - You can only use the question mark `?` wildcard character at the end of the URL.
  - The following wildcard syntax IS supported: `http://*.juniper.net`, `http://www.juniper.ne?`, `http://www.juniper.n??`.
  - The following wildcard syntax is NOT supported: `*juniper.net`, `www.juniper.ne?`, `http://*juniper.net`, `http://*`.
4. Configure a custom URL category list custom object using the URL pattern list you created. The following example adds the `urllist2` list to the `custurl2` custom object:

```
user@host# set security utm custom-objects custom-url-category custurl2 value
urllist2
```

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Example: Configuring Express Antivirus Feature Profiles (CLI)

After you have created your custom object, configure an antivirus feature profile:

1. Select and configure the engine type. Because you are configuring express antivirus, you select the `juniper-express-engine`: The following example sets the engine type to `juniper-express-engine`:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
```

2. Select a time interval for updating the pattern database. The default antivirus `pattern-update` interval is once a day. You can choose to leave this default as is or you can change it. You can also force a manual update, if necessary. The following example sets the update interval to 12:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
pattern-update interval 12
```



### NOTE:

The command for changing the URL for the pattern database is:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
pattern-update url http://...
```

Under most circumstances, you should not need to change the default URL.

---

3. Configure the device to notify a specified administrator when patterns are updated. The following example enables an e-mail notification with a custom message and a custom subject line:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
pattern-update email-notify admin-email administrator@juniper.net
custom-message "pattern file was updated" custom-message-subject "AV pattern
file updated"
```

4. Configure a profile for the Juniper-Express-Engine. The following example creates the junexprof1 profile:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine profile
junexprof1
```

5. Configure a list of fallback options as block or log-and-permit. In most cases, the default is to block. You can use the default settings or you can change them. The following example configures fallback options as block for the junexprof1 profile:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprof1 fallback-options content-size block
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprof1 fallback-options default block
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprof1 fallback-options engine-not-ready block
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprof1 fallback-options out-of-resources block
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprof1 fallback-options timeout block
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprof1 fallback-options too-many-requests block
```

6. Configure the notification options. You can configure notifications for both fallback blocking and fallback nonblocking actions and for virus detection. You configure a custom message for the fallback blocking action and send a notification. The following example configures the device to send the \*\*\*virus-found\*\*\* notification for blocked traffic:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprof1 notification-options fallback-block custom-message ***virus-found***
notify-mail-sender
```

7. Configure a notification for protocol-only virus detection and send a notification. The following example configures the protocol-only virus detection for the junexprof1 profile:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprof1 notification-options virus-detection type protocol-only
notify-mail-sender
```

8. Configure a custom subject line for the custom message notification for both the sender and the recipient. The following example configures the device to add "Antivirus Alert" to the message subject line:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
jjunexprof1 notification-options fallback-block custom-message-subject "Antivirus
Alert" notify-mail-sender
```

9. Configure content size parameters. The following example configures the device to perform a TCP payload content size check before the scan request is sent:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexproflscan-options content-size-limit 20000
```

10. Configure intelligent prescreening. It is either on or off. The following example enables intelligent prescreening for the junexprofl profile:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprofl scan-options intelligent-prescreening
```



**NOTE:** Intelligent prescreening is only intended for use with non-encoded traffic. It is not applicable for mail protocols (SMTP, POP3, IMAP) and HTTP POST.

The following example disables intelligent prescreening for the junexprofl profile:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprofl can-options no-intelligent-prescreening
```

11. Configure the time-out settings. The following example sets the scan-mode timeout to 1800 seconds:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprofl scan-options timeout 1800
```

12. Configure trickling settings. The following example indicates that if the device receives a packet within a 600 second period during a file transfer or while performing an antivirus scan, it should not timeout:

```
user@host# set security utm feature-profile anti-virus juniper-express-engine
junexprofl scan-options timeout 600
```



**NOTE:** Trickling applies only to HTTP.

13. Configure the antivirus scanner to use MIME bypass lists and exception lists. You can use your own custom object lists, or you can use the default list that ships with the device called junos-default-bypass-mime. The following examples enable the avmime2 and ex-avmime2 lists:

```
user@host# set security utm feature-profile anti-virus mime-whitelist list avmime2
user@host# set security utm feature-profile anti-virus mime-whitelist list avmime1
exception ex-avmime2
```

14. Configure the antivirus module to use URL bypass lists. If you are using a URL whitelist, this is a custom URL category you have previously configured as custom object. The following example enables the custurl1 bypass list:

```
user@host# set security utm feature-profile anti-virus url-whitelist custurl2
```

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Configuring Express Antivirus UTM Policies (CLI)

After you have created an antivirus feature profile, configure a UTM policy to which you can attach the feature profile:

1. Create a UTM policy for HTTP antivirus scanning. The following example creates the utmp3 policy:

```
user@host# set security utm utm-policy utmp3
```

2. Attach the policy to profile. The following example attaches the utmp3 policy to the junexprof1 HTTP profile:

```
user@host# set security utm utm-policy utmp3 anti-virus http-profile kasprofile1
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Attaching Express Antivirus UTM Policies to Security Policies (CLI)

After you create a UTM policy, create a security policy and attach the UTM policy to the security policy:

1. Enable and configure the security policy. The following example enables the p2 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p3 match
source-address any
user@host# set security policies from-zone trust to-zone untrust policy p3 match
destination-address any
user@host# set security policies from-zone trust to-zone untrust policy p3 match
application junos-http
```

2. Attach the UTM policy to the security policy. The following example attaches the utmp3 UTM policy to the p3 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p3 then
permit application-services utm-policy utmp3
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*





# Content Filtering

- Content Filtering Overview on page 665
- Content Filtering Protocol Support on page 666
- Example: Configuring Content Filtering on page 668
- Monitoring Content Filtering Configurations on page 677

## Content Filtering Overview

---

Content filtering blocks or permits certain types of traffic based on the MIME type, file extension, and protocol command. The content filter controls file transfers across the gateway by checking traffic against configured filter lists.

The content filter module evaluates traffic before all other UTM modules, except Web Filtering. Therefore, if traffic meets criteria configured in the content-filter, the content-filter acts first upon this traffic.

You can configure the following types of content filters:

- **MIME Pattern Filter** — MIME patterns are used to identify the type of traffic in HTTP and MAIL protocols. There are two lists of MIME patterns that are used by the content filter to determine the action to be taken. The block MIME list contains a list of MIME type traffic that is to be blocked by the content filter. The MIME exception list contains MIME patterns that are not to be blocked by the content filter and are generally subsets of items on the block list. Note that the exception list has a higher priority than the block list. If you have MIME entries that appear on both lists, those MIME types are not blocked by the content filter because the exception list takes priority. Therefore, when adding items to the exception list, it is to your advantage to be specific.
- **Block Extension List** — Because the name of a file is available during file transfers, using file extensions is a highly practical way to block or allow file transfers. The content filter list contains a list of file extensions to be blocked. All protocols support the use of the block extension list.
- **Protocol Command Block and Permit Lists** — Different protocols use different commands to communicate between servers and clients. By blocking or allowing certain commands, traffic can be controlled on the protocol command level.

The block and permit command lists are intended to be used in combination, with the permit list acting as an exception list to the block list.



NOTE: If a protocol command appears on the both the permit list and the block list, that command is permitted.

Because not all harmful files or components can be controlled by the MIME type or by the file extension, you can also use the content filter module to block ActiveX, Java Applets, and other types of content. The following types of content blocking are supported only for HTTP:

- Block ActiveX
- Block Java applets
- Block cookies
- Block EXE files
- Block ZIP files

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

---

## Content Filtering Protocol Support

- Understanding Content Filtering Protocol Support on page 666
- Specifying Content Filtering Protocols (CLI Procedure) on page 667

### Understanding Content Filtering Protocol Support

Each supported protocol may implement available content filters differently. Not all filtering capabilities are supported for each protocol.

This topic contains the following sections:

- HTTP Support on page 666
- FTP Support on page 667
- E-Mail Support on page 667

### HTTP Support

The HTTP protocol supports all content filtering features. With HTTP, the content filter remains in the gateway, checking every request and response between the HTTP client and server.

If an HTTP request is dropped due to content filtering, the client receives a response such as:

```
<custom drop message/user-configured drop
message>.<src_port><dst_ip>:<dst_port>Download request was dropped due to
<reason>
```

Therefore, a message may appear as follows:

```
Juniper Networks Firewall Content Filtering blocked request. 5.5.5.1:80->4.4.4.1:55247
Download request was dropped due to file extension block list
```

## FTP Support

The FTP protocol does not support all content filtering features. It supports only the following: Block Extension List and Protocol Command Block List.

When content filtering blocks an FTP request, the following response is sent through the control channel:

```
550 <src_ip>:<src_port>-<dst_ip>:<dst_port><custom drop message/user-configured
drop message> for Content Filtering file extension block list.>
```

Therefore, a message may appear as follows:

```
550 5.5.5.1:21->4.4.4.1:45237 Requested action not taken and the request is dropped for
Content Filtering file extension block list
```

## E-Mail Support

E-mail protocols (SMTP, IMAP, POP3) have limited content filtering support for the following features: Block Extension List, Protocol Command Block List, and MIME Pattern Filtering. Support is limited for e-mail protocols for the following reasons:

- The content filter scans only one level of an e-mail header. Therefore recursive e-mail headers and encrypted attachments are not scanned.
- If an entire e-mail is MIME encoded, the content filter can only scan for the MIME type.
- If any part of an e-mail is blocked due to content filtering, the original e-mail is dropped and replaced by a text file with an explanation for why the e-mail was blocked.

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Specifying Content Filtering Protocols (CLI Procedure)

To configure content filtering protocols, use the following CLI configuration statements:

```
content-filtering {
 profile name {
 permit-command cmd-list
 block-command cmd-list
 block-extension file-ext-list
 block-mime {
 list mime-list
 exception ex-mime-list
 }
 }
 block-content-type {
 activex
 java-applet
 exe
 zip
 http-cookie
```

```
 }
 notification-options {
 type { message }
 notify-mail-sender
 custom-message msg
 }
 }
 traceoptions {
 flag {
 all
 basic
 detail
 }
 }
}
```

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

---

## Example: Configuring Content Filtering

- Content Filtering Configuration Overview on page 668
- Example: Configuring Content Filtering Custom Objects on page 669
- Example: Configuring Content Filtering Feature Profiles on page 671
- Example: Configuring Content Filtering UTM Policies on page 674
- Example: Attaching Content Filtering UTM Policies to Security Policies on page 675

## Content Filtering Configuration Overview

Content security filter is a new feature that blocks or allows certain type of traffic base on the mime type, file extension, protocol commands and embedded object type. The content filter controls file transfers across the gateway by checking traffic against configured filter lists. The content filtering module evaluates traffic before all other UTM modules, if traffic meets the criteria configured in the content filter, the content filter acts first upon this traffic. The following procedure lists the recommended order in which you should configure content filters:

1. Configure UTM custom objects for the feature. See Example: Configuring Content Filtering Custom Objects.
2. Configure the main feature parameters using feature profiles. See Example: Configuring Content Filtering Feature Profiles.
3. Configure a UTM policy for each protocol and attach this policy to a profile. See Example: Configuring Content Filtering UTM Policies.
4. Attach the UTM policy to a security policy. See Example: Attaching Content Filtering UTM Policies to Security Policies.

**Related Topics**    • *Junos OS Feature Support Reference for SRX Series and J Series Devices*  
• Example: Configuring Content Filtering Custom Objects on page 669

- Example: Configuring Content Filtering Feature Profiles on page 671
- Example: Configuring Content Filtering UTM Policies on page 674
- Example: Attaching Content Filtering UTM Policies to Security Policies on page 675

## Example: Configuring Content Filtering Custom Objects

This example shows how to configure content filtering custom objects.

- Requirements on page 669
- Overview on page 669
- Configuration on page 669
- Verification on page 671

### Requirements

Before you begin:

1. Decide on the type of content filter you require. See “Content Filtering Overview” on page 665.
2. Understand the order in which content filtering parameters are configured. See “Content Filtering Configuration Overview” on page 668.

### Overview

In this example, you define custom objects that are used to create content filtering profiles. You perform the following tasks to define custom objects:

1. Create two protocol command lists called `ftpprotocolcom1` and `ftpprotocolcom2`, and add `user`, `pass`, `port`, and `type` commands to it.
2. Create a filename extension list called `extlist2`, and add the `.zip`, `.js`, and `.vbs` extensions to it.
3. Define block-mime list call `cfmime1` and add patterns to the list.

### Configuration

#### CLI Quick Configuration

To quickly configure content filtering custom objects, copy the following commands and paste them into the CLI.

```
[edit]
set security utm custom-objects protocol-command ftpprotocolcom1 value [user pass port
type]
set security utm custom-objects protocol-command ftpprotocolcom2 value [user pass port
type]
set security utm custom-objects filename-extension extlist2 value [zip js vbs]
set security utm custom-objects mime-pattern cfmime1 value [video/quicktime
image/x-portable-anymap x-world/x-vrml]
set security utm custom-objects mime-pattern ex-cfmime1 value
[video/quicktime-inappropriate]
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure content filtering custom objects:

1. Create two protocol command lists.

```
[edit security utm]
user@host# set custom-objects protocol-command ftpprotocom1
[edit security utm]
user@host# set custom-objects protocol-command ftpprotocom2
```

2. Add protocol commands to the list.

```
[edit security utm]
user@host# set custom-objects protocol-command ftpprotocom1 value [user pass
port type]
[edit security utm]
user@host# set custom-objects protocol-command ftpprotocom2 value [user pass
port type]
```

3. Create a filename extension list.

```
[edit security utm]
user@host# set custom-objects filename-extension extlist2
```

4. Add extensions to the list.

```
[edit security utm]
user@host# set custom-objects filename-extension extlist2 value [zip js vbs]
```

5. Create antivirus scanning lists.

```
[edit security utm]
user@host# set custom-objects mime-pattern cfmime1
user@host# set custom-objects mime-pattern ex-cfmime1
```

6. Add patterns to the lists.

```
[edit security utm]
user@host# set custom-objects mime-pattern cfmime1 value [video/quicktime
image/x-portable-anymap x-world/x-vrml]
user@host# set custom-objects mime-pattern ex-cfmime1 value
[video/quicktime-inappropriate]
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
userhost#show security utm
custom-objects {
 mime-pattern {
 cfmime1 {
 value [video/quicktime image/x-portable-anymap x-world/x-vrml];
 }
 ex-cfmime1 {
```

```

 value video/quicktime-inappropriate;
 }
}
filename-extension {
 extlist2 {
 value [zip js vbs];
 }
}
protocol-command {
 ftpprotocom1 {
 value [user pass port type];
 }
}
protocol-command {
 ftpprotocom2 {
 value [user pass port type];
 }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Content Filtering Custom Objects on page 671

### Verifying Content Filtering Custom Objects

**Purpose** Verify the content filtering custom objects.

**Action** From operational mode, enter the **show configuration security utm** command.

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Example: Configuring Content Filtering Feature Profiles

This example describes how to configure the content filtering feature profiles.

- Requirements on page 671
- Overview on page 672
- Configuration on page 672
- Verification on page 674

### Requirements

Before you begin:

1. Decide on the type of content filter you require. See “Content Filtering Overview” on page 665.
2. Create custom objects. See “Content Filtering Configuration Overview” on page 668.

## Overview

In this example, you configure a feature profile called `confilter1` and specify the following custom objects to be used for filtering content:

1. Apply the `ftpprotocol1` protocol command list custom object to `confilter1`.
2. Apply blocks to Java applets, executable files, and HTTP cookies.
3. Apply the extension list `extlist2` custom object to `confilter1` for blocking extensions.
4. Apply the MIME pattern list custom objects `cfmime1` and `ex-cfmime1` to the `confilter1` for blocking MIME types.
5. Apply the protocol permit command custom object `ftpprotocol2` to `confilter1`. (The permit protocol command list acts as an exception list for the block protocol command list.)



**NOTE:** Protocol command lists, both permit and block, are created by using the same custom object.

6. Configure a custom message to send a notification.

## Configuration

### CLI Quick Configuration

To quickly configure the content filtering feature profile, copy the following commands and paste them into the CLI.

```
[edit]
set security utm feature-profile content-filtering profile confilter1
set security utm feature-profile content-filtering profile confilter1 block-command
 ftpprotocol1
set security utm feature-profile content-filtering profile confilter1 block-content-type
 java-applet exe http-cookie
set security utm feature-profile content-filtering profile confilter1 block-extension extlist2
set security utm feature-profile content-filtering profile confilter1 block-mime list cfmime1
 exception ex-cfmime1
set security utm feature-profile content-filtering profile confilter1 permit-command
 ftpprotocol2
set security utm feature-profile content-filtering profile confilter1 notification-options
 custom-message "the action is not taken" notify-mail-sender type message
```

### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a content filtering feature profiles:

1. Create a content filtering profile.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1
```



2. Apply a protocol command list custom object to the profile.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1 block-command
ftpprotocol1
```

3. Apply blocks to available content.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1 block-content-type
java-applet exe http-cookie
```

4. Apply an extension list custom object to the profile.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1 block-extension
extlist2
```

5. Apply pattern list custom objects to the profile.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1 block-mime list
cfmime1 exception ex-cfmime1
```

6. Apply the protocol permit command custom object to the profile.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1 permit-command
ftpprotocol2
```

7. Configure the notification options.

```
[edit security utm]
user@host# set feature-profile content-filtering profile confilter1m
notification-options custom-message "the action is not taken" notify-mail-sender
type message
```

**Results** From configuration mode, confirm your configuration by entering the **show security utm** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security utm
feature-profile {
 content-filtering {
 profile contentfilter1;
 profile confilter1 {
 permit-command ftpprotocol2;
 block-command ftpprotocol1;
 block-extension extlist2;
 block-mime {
 list cfmime1;
 exception ex-cfmime1;
 }
 }
 block-content-type {
 java-applet;
 exe;
 http-cookie;
 }
 }
}
```

```
 }
 notification-options {
 type message;
 notify-mail-sender;
 custom-message " the action is not taken";
 }
 }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying the Configuration of Content Filtering Feature Profile on page 674

#### Verifying the Configuration of Content Filtering Feature Profile

**Purpose** Verify the content filtering feature profile.

**Action** From operational mode, enter the **show configuration security utm** command.

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Example: Configuring Content Filtering UTM Policies

This example describes how to create a content filtering UTM policy to attach to your feature profile.

- Requirements on page 674
- Overview on page 674
- Configuration on page 675
- Verification on page 675

### Requirements

Before you begin:

1. Decide on the type of content filter you require. See “Content Filtering Overview” on page 665.
2. Configure UTM custom objects for each feature and define the content-filtering profile. See “Content Filtering Configuration Overview” on page 668.

### Overview

You configure UTM policies to selectively enforce various UTM solutions on network traffic passing through a UTM-enabled device. Through feature profiles you associate custom objects to these policies and specify blocking or permitting certain types of traffic.

In this example, you configure a UTM policy called `utmp4`, and then assign the preconfigured feature profile `confilter1` to this policy.

## Configuration

### Step-by-Step Procedure

To configure a content filtering UTM policy:

You can configure different protocol applications in utm-policy. The example only show http and not other protocols. Earlier we configure custom objects for ftp (ftpprotocol1 and ftpprotocol2). We should add content filter policy for ftp. e.g.

```
set security utm utm-policy utmp4 content-filtering ftp upload-profile confilter1
```

```
set security utm utm-policy utmp4 content-filtering ftp download-profile confilter1
```

1. Create a UTM policy.

```
[edit security utm]
user@host# set utm-policy utmp4
```

2. Attach the UTM policy to the profile.

```
[edit security utm]
user@host# set utm-policy utmp4 content-filtering http-profile contentfilter1
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

## Verification

To verify the configuration is working properly, enter the **show security utm** command.

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Example: Attaching Content Filtering UTM Policies to Security Policies

This example shows how to create a security policy and attach the UTM policy to the security policy.

- Requirements on page 675
- Overview on page 676
- Configuration on page 676
- Verification on page 677

## Requirements

Before you begin:

1. Configure UTM custom objects, define the content filtering profile, and create a UTM policy. See “Content Filtering Configuration Overview” on page 668.
2. Enable and configure a security policy. See “Example: Configuring a Security Policy to Permit or Deny All Traffic” on page 121.

## Overview

By attaching content filtering UTM policies to security policies, you can filter traffic transiting from one security zone to another.

In this example, you create a security policy called p4 and specify that traffic from any source address to any destination address with an HTTP application matches the criteria. You then assign a UTM policy called utmp4 to the security policy p4. This UTM policy applies to any traffic that matches the criteria specified in the security policy p4.

## Configuration

**CLI Quick Configuration** To quickly attach a content filtering UTM policy to a security policy, copy the following commands and paste them into the CLI.

```
[edit]
set security policies from-zone trust to-zone untrust policy p4 match source-address any
set security policies from-zone trust to-zone untrust policy p4 match destination-address
 any
set security policies from-zone trust to-zone untrust policy p4 match application
 junos-http
set security from-zone trust to-zone untrust policy p4 then permit application-services
 utm-policy utmp4
```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To attach a UTM policy to a security policy:

1. Create a security policy.

```
[edit]
user@host# edit security policies from-zone trust to-zone untrust policy p4
```

2. Specify the match conditions for the policy.

```
[edit security policies from-zone trust to-zone untrust policy p4]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application junos-http
```

3. Attach the UTM policy to the security policy.

```
[edit security policies from-zone trust to-zone untrust policy p4]
user@host# set then permit application-services utm-policy utmp4
```

**Results** From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security policies
 from-zone trust to-zone untrust {
 policy p4 {
```

```

match {
 source-address any;
 destination-address any;
 application junos-http;
}
then {
 permit {
 application-services {
 utm-policy utmp4;
 }
 }
}
}
}
default-policy {
 permit-all;
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform this task:

- Verifying Attaching Content Filtering UTM Policies to Security Policies on page 677

### Verifying Attaching Content Filtering UTM Policies to Security Policies

**Purpose** Verify the attachment of the content filtering UTM policy to the security policy.

**Action** From operational mode, enter the **show security policy** command.

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Monitoring Content Filtering Configurations

**Purpose** View content filtering statistics.

**Action** To view content filtering statistics in the CLI, enter the **user@host > show security utm content-filtering statistics** command.

The content filtering show statistics command displays the following information:

```

Base on command list: # Blocked
Base on mime list: # Blocked
Base on extension list: # Blocked
ActiveX plugin: # Blocked
Java applet: # Blocked
EXE files: # Blocked
ZIP files: # Blocked
HTTP cookie: # Blocked

```

To view content filtering statistics using J-Web:

1. Select **Monitor>Security>UTM>Content Filtering**.

The following statistics becomes viewable in the right pane.

Base on command list: # Passed # Blocked  
Base on mime list: # Passed # Blocked  
Base on extension list: # Passed # Blocked  
ActiveX plugin: # Passed # Blocked  
Java applet: # Passed # Blocked  
EXE files: # Passed # Blocked  
ZIP files: # Passed # Blocked  
HTTP cookie: # Passed # Blocked

2. You can click **Clear Content filtering statistics** to clear all current viewable statistics and begin collecting new statistics.

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

# Web Filtering

- Web Filtering Overview on page 679
- Integrated Web Filtering on page 680
- Redirect Web Filtering on page 691
- Local Web Filtering on page 700
- Monitoring Web Filtering Configurations on page 704

## Web Filtering Overview

---

Web filtering lets you manage Internet usage by preventing access to inappropriate Web content. There are three types of Web filtering solutions:

- Integrated Web filtering—The integrated Web filtering solution intercepts every HTTP request in a TCP connection. In this case, the decision making is done on the device after it identifies the category for a URL either from user-defined categories or from a category server (SurfControl Content Portal Authority provided by Websense).



NOTE: The integrated Web filtering feature is a separately licensed subscription service. When the license key for Web filtering has expired, no URLs are sent to the category server for checking, only local user-defined categories are checked.

- Redirect Web filtering—The redirect Web filtering solution intercepts HTTP requests and sends them to an external URL filtering server, provided by Websense, to determine whether to block the requests.



NOTE: Redirect Web filtering does not require a license.

- Local Web filtering—The local Web filtering solution intercepts every HTTP request in a TCP connection. In this case, the decision making is done on the device after it looks up a URL to determine if it is in the whitelist or blacklist based on its user-defined category.



NOTE: Local Web filtering does not require a license or a remote category server.

You can bind either Web filtering profiles or antivirus profiles, or both, to a firewall policy. When both are bound to a firewall policy, Web filtering is applied first, then antivirus is applied. If a URL is blocked by Web filtering, the TCP connection is closed and no antivirus scanning is necessary. If a URL is permitted, the content of the transaction is then passed to the antivirus scanning process.

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

---

## Integrated Web Filtering

- Understanding Integrated Web Filtering on page 680
- Integrated Web Filtering Configuration Overview on page 682
- Configuring Integrated Web Filtering (J-Web Procedure) on page 683
- Example: Configuring Integrated Web Filtering (CLI) on page 687
- Displaying Global SurfControl URL categories on page 691

### Understanding Integrated Web Filtering

With integrated Web filtering, the firewall intercepts every HTTP request in a TCP connection and extracts the URL from the HTTP request. Each individual HTTP request is blocked or permitted based on URL filtering profiles defined by you. The decision making is done on the device after it identifies a category for a URL.

A URL category is a list of URLs grouped by content. URL categories are predefined and maintained by SurfControl or are defined by you. SurfControl maintains about 40 predefined categories. When defining your own URL categories, you can group URLs and create categories specific to your needs.

You define your own categories using URL pattern list and custom URL category list custom objects. Once defined, you can select your categories when you configure your Web filtering profile. Each category can have a maximum of 20 URLs. When you create a category, you can add either the URL or the IP address of a site. When you add a URL to a user-defined category, the device performs DNS lookup, resolves the host name into IP addresses, and caches this information. When a user tries to access a site with the IP address of the site, the device checks the cached list of IP addresses and tries to resolve the hostname. Many sites have dynamic IP addresses, meaning that their IP addresses change periodically. A user attempting to access a site can type an IP address that is not in the cached list on the device. Therefore, if you know the IP addresses of sites you are adding to a category, enter both the URL and the IP address(es) of the site.



**NOTE:** If a URL appears in both a user-defined category and a predefined category, the device matches the URL to the user-defined category.

---



**NOTE:** Web filtering is performed on all the methods defined in HTTP 1.0 and HTTP 1.1.

---



This topic contains the following sections:

- Integrated Web Filtering Process on page 681
- Integrated Web Filtering Cache on page 681
- Integrated Web Filtering Profiles on page 681
- Profile Matching Precedence on page 682

## Integrated Web Filtering Process

This is a general description of how Web traffic is intercepted and acted upon by the Web filtering module.

1. The device intercepts a TCP connection.
2. The device intercepts each HTTP request in the TCP connection.
3. The device extracts each URL in the HTTP request and checks its URL filter cache.
4. Global Web filtering white and blacklists are checked first for block or permit.
5. If the HTTP request URL is allowed based on cached parameters, it is forwarded to the webserver. If there is no cache match, a request for categorization is sent to the SurfControl server. (If the HTTP request URL is blocked, the request is not forwarded and a notification message is logged.)
6. In the allowed case, the SurfControl server responds with the corresponding category.
7. Based on the identified category, if the URL is permitted, the device forwards the HTTP request to the webserver. If the URL is not permitted, then a deny page is sent to the HTTP client.

## Integrated Web Filtering Cache

By default, the device retrieves and caches the URL categories from the SurfControl CPA server. This process reduces the overhead of accessing the SurfControl CPA server each time the device receives a new request for previously requested URLs. You can configure the size and duration of the cache, according to the performance and memory requirements of your networking environment. The lifetime of cached items is configurable between 1 and 1800 seconds with a default value of 300 seconds.



**NOTE:** Caches are not preserved across device reboots or power losses.

## Integrated Web Filtering Profiles

You configure Web filtering profiles that permit or block URLs according to defined categories. A Web filtering profile consists of a group of URL categories assigned one of the following actions:

- Permit — The device always allows access to the websites in this category.
- Block — The device blocks access to the websites in this category. When the device blocks access to this category of websites, it displays a message in your browser indicating the URL category.

- Blacklist — The device always blocks access to the websites in this list. You can create a user-defined category or use a predefined category.
- Whitelist — The device always allows access to the websites in this list. You can create a user-defined category or use a predefined category.



**NOTE:** A predefined profile is provided and can be used if you choose not to define your own profile.

A Web filtering profile may contain one blacklist or one whitelist, multiple user-defined and/or predefined categories each with a permit or block action, and an *Other* category with a permit or block action. You can define an action for all *Other* categories in a profile to specify what to do when the incoming URL does not belong to any of the categories defined in the profile. If the action for the *Other* category is block, the incoming URL is blocked if it does not match any of the categories explicitly defined in the profile. If an action for the *Other* category is not specified, the default action of permit is applied to the incoming URL not matching any category.

## Profile Matching Precedence

When a profile employs several categories for URL matching, those categories are checked for matches in the following order:

1. If present, the global blacklist is checked first. If a match is made, the URL is blocked. If no match is found...
2. The global whitelist is checked next. If a match is made, the URL is permitted. If no match is found...
3. User-defined categories are checked next. If a match is made, the URL is blocked or permitted as specified. If no match is found...
4. Predefined categories are checked next. If a match is made, the URL is blocked or permitted as specified. If no match is found...
5. The Other category is checked next. If a match is made, the URL is blocked or permitted as specified.

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Integrated Web Filtering Configuration Overview

When configuring Web filtering, if you are using custom objects, you must first create those. Custom objects may include URL Pattern Lists and Custom URL Category Lists. Once you have created your custom objects, you can configure Web filtering.

For each UTM feature, you should configure feature parameters in the following order:

1. Configure UTM custom objects for the feature. The following example creates the url-pattern and custom-url-category custom objects:

```
user@host# set security utm custom-objects url-pattern
```

```
user@host# set security utm custom-objects custom-url-category
```

2. Configure the main feature parameters using feature profiles. The following example creates the Web-filtering feature profile:

```
user@host# set security utm feature-profile Web-filtering
```

3. Configure a UTM policy for each protocol and attach this policy to a profile. The following example attaches the utmp5 UTM policy to the webhttp1 profile:

```
user@host# set security utm utm-policy utmp5 web-filtering http-profile webhttp1
```

4. Attach the UTM policy to a security policy. The following example attaches the utmp5 UTM policy to the p5 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p5 then
permit application-services utm-policy utmp5
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Configuring Integrated Web Filtering (J-Web Procedure)

- Configuring Integrated Web Filtering Custom Objects (J-Web Procedure) on page 683
- Configuring Integrated Web Filtering Feature Profiles (J-Web Procedure) on page 684
- Configuring Integrated Web Filtering UTM Policies (J-Web Procedure) on page 686
- Attaching Integrated Web Filtering UTM Policies to Security Policies (J-Web Procedure) on page 687

### Configuring Integrated Web Filtering Custom Objects (J-Web Procedure)

To configure Web filtering using the J-Web Configuration editor, if you are using custom objects, you must first create those custom objects (URL pattern list, custom URL category list).



**NOTE:** Rather than or in addition to custom object lists, you can use included default lists and included whitelist and blacklist categories.

Configure a URL pattern list custom object and a URL category list custom object:

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the URL Pattern List tab, click **Add** to create URL pattern lists.
3. Next to **URL Pattern Name**, enter a unique name. This name appears in the Custom URL Category List Custom Object page for selection.
4. Next to **URL Pattern Value**, enter the URL or IP address you want added to list for bypassing scanning.

When entering the URL pattern, note the following wildcard character support:

- The `\*\.[\]\?*` wildcard characters are supported.
- You must precede all wildcard URLs with `http://`.

- You can only use the asterisk \* wildcard character if it is at the beginning of the URL and is followed by a period.
  - You can only use the question mark ? wildcard character at the end of the URL.
  - The following wildcard syntax IS supported: **http://\*.juniper.net**, **http://www.juniper.ne?**, **http://www.juniper.n??**.
  - The following wildcard syntax is NOT supported: **\*juniper.net** , **www.juniper.ne?**, **http://\*juniper.net**, **http://\***.
5. Click **Add** to add your URL pattern to the Values list box. The list can contain up to 8192 items. You can also select an entry and use the Delete button to delete it from the list. Continue to add URLs or IP addresses in this manner.
  6. Click **OK** to save the selected values as part of the URL pattern list you have created.
  7. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure a custom URL category list.

8. Select **Configure>Security>UTM>Custom Objects**.
9. From the URL Category List tab, click **Add** to create URL category lists.
10. Next to **URL Category Name**, enter a unique name. This name appears in the URL Whitelist, Blacklist, and Custom Category lists when you configure Web filtering global options.
11. In the **Available Values** box, select a URL Pattern List name from the list for bypassing scanning and click the right arrow button to move it to the Selected Values box.
12. Click **OK** to check your configuration and save the selected values as part of the custom URL list you have created, then click **Commit Options>Commit**.
13. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

### Configuring Integrated Web Filtering Feature Profiles (J-Web Procedure)

After you create custom objects, configure the integrated Web filtering feature profile:

1. Select **Configure>Security>UTM>Global options**.
2. In the Web Filtering section, next to **URL whitelist**, select the custom URL list you created from the available options. This is the first filtering category that both integrated and redirect Web filtering use. If there is no match, the URL is sent to the SurfControl server.

3. Next to URL blacklist, from the **Custom URL** list, select the list that you created. This is the first filtering category that both integrated and redirect Web filtering use. If there is no match, the URL is sent to the SurfControl server.
4. In the Filtering Type section, select the type of Web filtering engine you are using. In this case, you would select **Surf Control Integrated**.
5. In the SurfControl Integrated options section, next to **Cache timeout**, enter a timeout limit, in minutes, for expiring cache entries (24 hours is the default and the maximum allowed life span).
6. Next to **Cache Size**, enter a size limit, in kilobytes, for the cache (500 KB is the default).
7. Next to **Server Host**, enter the Surf Control server name or IP address.
8. Next to **Server Port**, enter the port number for communicating with the Surf Control server (default ports are 80, 8080, and 8081).
9. Click **OK** to save these values.
10. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.
11. Under UTM, in the left pane, select **Web Filtering**.
12. In Web Filtering Profiles Configuration, click **Add** to create a profile for Surf Control Integrated Web filtering. (To edit an existing item, select it and click **Edit**.)
13. In **Profile name**, enter a unique name.
14. From the Profile Type list, select **Surf Control**.
15. Next to Default action, select **Permit, Log and permit**, or **Block**. This is the default action for this profile for requests that experience errors.
16. Next to **Custom Block Message**, enter a custom message to be sent when HTTP requests are blocked.
17. Next to **Timeout**, enter a value, in seconds. Once this limit is reached, fail mode settings are applied. The default here is 10 seconds. You can enter a value from 10 to 240 seconds.
18. Next to **Custom block message subject**, enter text to appear in the subject line of your custom message for this block notification.
19. Select the **Fallback options** tab.
20. Next to Default, select **Log and Permit** or **Block** as the action to occur when a request fails for any reason not specifically called out.
21. Next to Server Connectivity, select **Log and Permit** or **Block** as the action to occur when a request fails for this reason.
22. Next to Timeout, select **Log and Permit** or **Block** as the action to occur when a request fails for this reason.

23. Next to Too Many Requests, select **Log and Permit** or **Block** as the action to occur when a request fails for this reason.
24. Click **Save**.
25. Under UTM, in the left pane, select **Custom Objects**.
26. Select the **URL category list** tab.
27. In the custom URL category list section, click **Add** to use a configured custom URL category list custom object in the profile.
28. Next to **Categories**, select a configured custom object from the list.
29. Next to Actions, select **Permit**, **Block**, or **Log and Permit** from the list.
30. Click **Add**.
31. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
32. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

### Configuring Integrated Web Filtering UTM Policies (J-Web Procedure)

After you have created a content filtering feature profile, configure a UTM policy for Web filtering:

1. Select **Configure>Security>Policy>UTM Policies**.
2. From the UTM policy configuration window, click **Add** to configure a UTM policy. The policy configuration pop-up window appears.
3. Select the **Main** tab.
4. In the **Policy Name** box, enter a unique name.
5. In the **Session per client limit** box, enter a session per client limit from 0 to 20000 for this UTM policy.
6. For Session per client over limit, select one of the following: **Log and Permit** or **Block**. This is the action the device takes when the session per client limit for this UTM policy is exceeded.
7. Select the **Web Filtering profiles** tab.
8. Next to **HTTP profile**, select the profile you have configured from the list.
9. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
10. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

### Attaching Integrated Web Filtering UTM Policies to Security Policies (J-Web Procedure)

After you have created a UTM policy, attach it to a security policy:

1. Select **Configure>Security>Policy>FW Policies**.
2. From the Security Policy window, click **Add** to configure a security policy with UTM. The policy configuration pop-up window appears.
3. In the Policy tab, enter a name in the **Policy Name** box.
4. Next to **From Zone**, select a zone from the list.
5. Next to **To Zone**, select a zone from the list.
6. Choose a **Source Address**.
7. Choose a **Destination Address**.
8. Choose an application by selecting **junos-protocol** (for all protocols that support Web filtering, http in this case) in the Application Sets box and clicking the —> button to move it to the Matched box.
9. Next to Policy Action, select one of the following: **Permit**, **Deny**, or **Reject**.



**NOTE:** When you select Permit for Policy Action, several additional fields become available in the Applications Services tab, including UTM Policy.

10. Select the **Application Services** tab.
11. Next to **UTM Policy**, select the appropriate policy from the list. This attaches your UTM policy to the security policy.
12. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
13. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

You must activate your policy in order to apply it.

### Example: Configuring Integrated Web Filtering (CLI)

- Example: Configuring Integrated Web Filtering Custom Objects (CLI) on page 687
- Example: Configuring Integrated Web Filtering Feature Profiles (CLI) on page 689
- Example: Configuring Integrated Web Filtering UTM Policies (CLI) on page 690
- Example: Attaching Integrated Web Filtering UTM Policies to Security Policies (CLI) on page 691

### Example: Configuring Integrated Web Filtering Custom Objects (CLI)

To configure integrated Web filtering using the CLI, you must first create your custom objects.

To create custom objects:

1. Create a URL pattern list and add values to it. The following example creates the `urllist3` custom object:

```
user@host# set security utm custom-objects url-pattern urllist3 value
[http://www.juniper.net 1.2.3.4]
```

When entering the URL pattern, note the following wildcard character support:

- The `\*\.[]\?` wildcard characters are supported.
- You must precede all wildcard URLs with `http://`.
- You can only use the asterisk `*` wildcard character if it is at the beginning of the URL and is followed by a period.
- You can only use the question mark `?` wildcard character at the end of the URL.
- The following wildcard syntax IS supported: `http://*.juniper.net`, `http://www.juniper.ne?`, `http://www.juniper.n??`.
- The following wildcard syntax is NOT supported: `*juniper.net`, `www.juniper.ne?`, `http://*juniper.net`, `http://*`.



**NOTE:** Rather than or in addition to custom object lists, you can use included default lists and included whitelist and blacklist categories.

---



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure custom URL category lists.

---

2. Configure a custom URL category list custom object using a URL pattern list. The following example applies the `custurl3` URL pattern list to the `custurl3` custom object:

```
user@host# set security utm custom-objects custom-url-category custurl3 value
urllist3
```

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*



### Example: Configuring Integrated Web Filtering Feature Profiles (CLI)

After you create your custom objects, configure the surf-control-integrated Web filtering feature profile:

1. If you are using included global whitelist and blacklist categories, select those global categories. This is the first filtering category that both integrated and redirect Web filtering use. If no match is made, the URL is forwarded to the SurfControl server. The following example selects the Drugs\_Alcohol\_Tobacco blacklist and the Computing\_Internet whitelist:

```
user@host# set security utm feature-profile web-filtering url-blacklist
Drugs_Alcohol_Tobacco
user@host# set security utm feature-profile web-filtering url-whitelist
Computing_Internet
```

2. Select surf-control-integrated as your Web filtering engine:

```
user@host# set security utm feature-profile web-filtering surf-control-integrated
```

3. Set the cache size parameters for surf-control-integrated Web filtering (500 KB is the default). The following example sets the cache size to 500:

```
user@host# set security utm feature-profile web-filtering surf-control-integrated
cache size 500
```

4. Set the cache timeout parameters for surf-control-integrated Web filtering (24 hours is the default and the maximum allowed life span). The following example sets the timeout to 1800:

```
user@host# set security utm feature-profile web-filtering surf-control-integrated
cache timeout 1800
```

5. Set the Surf Control server name or IP address. The following example sets the surfcontrolserver hostname:

```
user@host# set security utm feature-profile web-filtering surf-control-integrated
server host surfcontrolserver
```

6. Enter the port number for communicating with the Surf Control server. (Default ports are 80, 8080, and 8081.) The following example sets the port number to 8080:

```
user@host# set security utm feature-profile web-filtering surf-control-integrated
server port 8080
```

7. Create a surf-control-integrated profile name. The following example creates the surfprofile1 profile:

```
user@host# set security utm feature-profile web-filtering surf-control-integrated
profile surfprofile1
```

8. Select a category from the included whitelist and blacklist categories or select a custom URL category list you created for filtering against. Then enter an action (permit, log and permit, block) to go with the filter as follows (do this as many time as necessary to compile your whitelists and blacklists and their accompanying actions). The following example blocks URLs in the custurl2 category:

```
user@host# set security utm feature-profile web-filtering surf-control-integrated
profile surfprofile1 category custurl2 action block
```

9. Enter a custom message to be sent when HTTP requests are blocked. The following example configures the device to send an **\*\*\*access denied\*\*\*** message:

```
user@host# set security utm feature-profile web-filtering surf-control-integrated
profile surfprofile1 custom-block-message ***access denied***
```

10. Select a default action (permit, log and permit, block) for this profile for requests that experience errors. The following example sets the default action to block:

```
user@host# set security utm feature-profile web-filtering surf-control-integrated
profile surfprofile1 default block
```

11. Select fallback settings (block or log and permit) for this profile. The fallback actions are taken when errors in each configured category occur. The following example sets fallback settings to block:

```
user@host# set security utm feature-profile web-filtering surf-control-integrated
profile surfprofile1 fallback-settings default block
user@host# set security utm feature-profile web-filtering surf-control-integrated
profile surfprofile1 fallback-settings server-connectivity block
user@host# set security utm feature-profile web-filtering surf-control-integrated
profile surfprofile1 fallback-settings timeout block
user@host# set security utm feature-profile web-filtering surf-control-integrated
profile surfprofile1 fallback-settings too-many-requests block
```

12. Enter a timeout value, in seconds. Once this limit is reached, fail mode settings are applied. The default here is 10 seconds. You can enter a value from 10 to 240 seconds. The following example sets the timeout value to 10:

```
user@host# set security utm feature-profile web-filtering surf-control-integrated
profile surfprofile1 timeout 10
```

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Configuring Integrated Web Filtering UTM Policies (CLI)

To configure a UTM policy:

1. Create the UTM policy. The following example creates the utmp5 policy:

```
user@host# set security utm utm-policy utmp5
```

2. Attach the UTM policy to a profile. The following example attaches the utmp5 policy to the surfprofile1 profile:

```
user@host# set security utm utm-policy utmp5 web-filtering http-profile surfprofile1
```

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Attaching Integrated Web Filtering UTM Policies to Security Policies (CLI)

To attach a UTM policy to a security policy:

1. Create and configure the security policy. The following example creates the p5 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p5 match
source-address any
user@host# set security policies from-zone trust to-zone untrust policy p5 match
destination-address any
user@host# set security policies from-zone trust to-zone untrust policy p5 match
application junos-http
```

2. Attach the UTM policy to the security policy. The following example attaches the utmp5 UTM policy to the p5 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p5 then
permit application-services utm-policy utmp5
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Displaying Global SurfControl URL categories

**Purpose** View global URL categories defined and maintained by SurfControl.

**Action** Enter the `user@host# show groups junos-defaults` CLI command. You can also look for `custom-url-category`.

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Redirect Web Filtering

- Understanding Redirect Web Filtering on page 691
- Redirect Web Filtering Configuration Overview on page 692
- Configuring Redirect Web Filtering (J-Web Procedure) on page 693
- Example: Configuring Redirect Web Filtering (CLI) on page 697

### Understanding Redirect Web Filtering

With redirect Web filtering, the Web filtering module intercepts an HTTP request. The URL in the request is then sent to the external Websense server, which makes a permit or a deny decision. If access is permitted to the URL in question, the original HTTP request and all the subsequent requests are sent to the intended HTTP server. But if access is denied to the URL in question, a blocking message is sent to the client.

This is a general description of how Web traffic is intercepted, redirected, and acted upon by the Web filtering module:

1. A Web client establishes a TCP connection with the webserver.
2. The Web client then sends an HTTP request.
3. The device intercepts the requests and extract URL. The URL is checked against Global Web filtering white and blacklists. If no match is made, the Websense server configuration parameters are utilized. Otherwise go to step 6.
4. The URL is sent to the Websense server for checking,
5. The Websense server returns a response indicating whether or not the URL is to be permitted or blocked.
6. If access is allowed, then the original HTTP request is sent to the webserver. If access is denied, the device sends a blocking message to the client and tears down the TCP connection.



NOTE: Web filtering is performed on all the methods defined in HTTP 1.0 and HTTP 1.1. However, redirect Web filtering does not support HTTPS traffic because it cannot be decrypted to obtain the URL.



NOTE: Decision making from real-time options provides a higher level of accuracy, therefore caching for redirect Web filtering is not supported.



NOTE: Redirect Web filtering does not require a subscription license.

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Redirect Web Filtering Configuration Overview

When configuring Web filtering, if you are using custom objects, you must first create those. Custom objects may include URL pattern lists and custom URL category lists. Once you have created your custom objects, you can configure Web filtering.

For each UTM feature, you should configure feature parameters in the following order:

1. First configure UTM custom objects for the feature. The following example creates the url-pattern and custom-url-category custom objects:

```
user@host# set security utm custom-objects url-pattern
user@host# set security utm custom-objects custom-url-category
```
2. Configure the main feature parameters using feature profiles. The following example creates the Web-filtering feature profile:

```
user@host# set security utm feature-profile Web-filtering
```

3. Configure a UTM policy for each protocol and attach this policy to a profile. The following example attaches the utmp6 UTM policy to the webhttp2 profile:

```
user@host# set security utm utm-policy utmp6 web-filtering http-profile webhttp2
```

4. Attach the UTM policy to a security policy. The following example attaches the utmp6 UTM policy to the p6 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p6 then
permit application-services utm-policy utmp6
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Configuring Redirect Web Filtering (J-Web Procedure)

- Configuring Redirect Web Filtering Custom Objects (J-Web Procedure) on page 693
- Configuring Redirect Web Filtering Feature Profiles (J-Web Procedure) on page 695
- Configuring Redirect Web Filtering UTM Policies (J-Web Procedure) on page 696
- Attaching Redirect Web Filtering UTM Policies to Security Policies (J-Web Procedure) on page 696

### Configuring Redirect Web Filtering Custom Objects (J-Web Procedure)

To configure Web filtering using the J-Web configuration editor, if you are using custom objects, you must first create those custom objects. (URL pattern list, custom URL category list).



**NOTE:** Rather than or in addition to custom object lists, you can use included default lists and included whitelist and blacklist categories. See “Integrated Web Filtering Profiles” on page 681 for profile list information.

Configure a URL Pattern List Custom Object as follows:



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure a custom URL category list.

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the **URL Pattern List** tab, click **Add** to create URL pattern lists.
3. Next to URL Pattern Name, enter a unique name for the list you are creating. This name appears in the Custom URL Category List Custom Object page for selection.
4. Next to URL Pattern Value, enter the URL or IP address you want added to list for bypassing scanning.



**NOTE:** URL pattern wildcard support—The wildcard rule is as follows: `\*\.[\]\?*` and you must precede all wildcard URLs with `http://`. You can only use “\*” if it is at the beginning of the URL and is followed by a “.”. You can only use “?” at the end of the URL. The following wildcard syntax IS supported: `http://*.juniper.net`, `http://www.juniper.net?`, `http://www.juniper.n??`. The following wildcard syntax is NOT supported: `*juniper.net`, `www.juniper.net?`, `http://*juniper.net`, `http://*`.

5. Click **Add** to add your URL pattern to the Values list box.

The list can contain up to 8192 items. You can also select an entry and use the Delete button to delete it from the list. Continue to add URLs or IP addresses in this manner.

6. Click **OK** to check your configuration and save the selected values as part of the URL pattern list you have created, then click **Commit Options>Commit**.
7. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

Configure a custom URL category list custom object as follows (see “Understanding URL Whitelists” on page 619 for overview information on URL whitelists):



**NOTE:** Because you use URL Pattern Lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure a custom URL category list.

1. Select **Configure>Security>UTM>Custom Objects**.
2. From the **URL Category List** tab, click **Add** to create URL category lists.
3. Next to URL Category Name, enter a unique name for the list you are creating. This name appears in the URL Whitelist, Blacklist, and Custom Category lists when you configure Web filtering global options.
4. In the Available Values box, select a URL Pattern List name from the list for bypassing scanning and click the right arrow button to move it to the Selected Values box.
5. Click **Add** to add your entry to the Values list box.

Within this box, you can select an entry and use the up and down arrows to change the order of the list. You can also select an entry and use the X button to delete it from the list. Continue to add URLs or IP addresses in this manner.

6. Click **OK** to check your configuration and save the selected values as part of the URL category list you have created, then click **Commit Options>Commit**.
7. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

### Configuring Redirect Web Filtering Feature Profiles (J-Web Procedure)

Now that your custom objects have been created, you can configure the redirect Web filtering feature profile.

1. Select **Configure>Security>UTM>Global options**.
2. In the **Web Filtering** tab, next to URL whitelist, select the Custom URL list you created from the available options.  
  
This is the first filtering category that both integrated and redirect Web filtering use. If there is no match, the URL is sent to the Websense server.
3. Next to URL blacklist, select the Custom URL list you created from the list. This is the first filtering category that both integrated and redirect Web filtering use. If there is no match, the URL is sent to the Websense server.
4. In the Filtering Type section, select the type of Web filtering engine you are using.  
  
In this case, you would select **Websense Redirect**.
5. Click **OK** to save these values.
6. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.
7. Select **Web Filtering**, under UTM, in the left pane.
8. Click **Add** to create a profile for redirect Web filtering. (To edit an existing item, select it and click the **Edit** button.)
9. In the **Main** tab, next to Profile name, enter a unique name for this Web filtering profile.
10. Select the Profile Type. In this case, select **Websense**.
11. Next to Account, enter the user account for which this profile is intended.
12. Next to Server, enter the Websense server name.
13. Next to Port, enter the port number for communicating with the Websense server (default ports are 80, 8080, and 8081).
14. Next to Sockets, enter the number of sockets used for communicating between the client and server (the default here is 1).
15. Next to Timeout, enter a timeout limit for requests.  
  
Once this limit is reached, fail mode settings are applied. The default here is 10 seconds. You can enter a value from 10 to 240 seconds.
16. Next to Custom Block Message, enter a custom message to be sent when HTTP requests are blocked.
17. Select the **Fallback options** tab.
18. Next to Default Action, select **Log and permit** or **Block**.

19. Next to Server Connectivity, select **Log and permit** or **Block** as the action to occur when a request fails for this reason.
20. Next to Timeout, select **Log and permit** or **Block** as the action to occur when a request fails for this reason.
21. Next to Too Many Requests, select **Log and permit** or **Block** as the action to occur when a request fails for this reason.
22. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
23. If the configuration item is saved successfully, you receive a confirmation and you must click **OK** again. If it is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

### Configuring Redirect Web Filtering UTM Policies (J-Web Procedure)

Next, you configure a UTM policy for Web filtering to which you attach the content filtering profile you have configured.

1. Select **Configure>Security>Policy>UTM Policies**.
2. From the UTM policy configuration window, click **Add** to configure a UTM policy.  
The policy configuration pop-up window appears.
3. Select the **Main** tab in the pop-up window.
4. In the Policy Name box, enter a unique name for the UTM policy you are creating.
5. In the Session per client limit box, enter a session per client limit from 0 to 20000 for this UTM policy.
6. For Session per client over limit, select one of the following: **Log and Permit** or **Block**.  
This is the action the device takes when the session per client limit for this UTM policy is exceeded.
7. Select the **Web Filtering profiles** tab in the pop-up window.
8. Next to HTTP profile, select the profile you have configured from the list.
9. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
10. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.

### Attaching Redirect Web Filtering UTM Policies to Security Policies (J-Web Procedure)

Next, you attach the UTM policy to a security policy that you create.

1. Select **Configure>Security>Policy>FW Policies**.
2. From the Security Policy window, click **Add** to configure a security policy with UTM.  
This takes you to the policy configuration pop-up window.



3. In the **Policy** tab, enter a name in the Policy Name box.
4. Next to From Zone, select a zone from the list.
5. Next to To Zone, select a zone from the list.
6. Choose a **Source Address**.
7. Choose a **Destination Address**.
8. Choose an Application. Do this by selecting junos-<protocol> (for all protocols that support Web filtering, http in this case) in the Application Sets box and clicking the —> button to move them to the Matched box.
9. Next to Default Policy Action, select one of the following: **Deny-All** or **Permit-All**.



**NOTE:** When you select Permit for Policy Action, several additional fields become available in the Applications Services tab, including UTM Policy.

10. Select the **Application Services** tab in the pop-up window.
  11. Next to UTM Policy, select the appropriate policy from the list.
- This attaches your UTM policy to the security policy.



**NOTE:** There are several fields on this page that are not described in this section. See the section on Security Policies for detailed information on configuring security policies and all the available fields.

12. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.
  13. If the policy is saved successfully, you receive a confirmation and you must click **OK** again. If the profile is not saved successfully, you can click **Details** in the pop-up window that appears to discover why.
- You must activate your policy in order to apply it.

### Example: Configuring Redirect Web Filtering (CLI)

- Example: Configuring Redirect Web Filtering Custom Objects (CLI) on page 697
- Example: Configuring Redirect Web Filtering Feature Profiles (CLI) on page 698
- Example: Configuring Redirect Web Filtering UTM Policies (CLI) on page 699
- Example: Attaching Redirect Web Filtering UTM Policies to Security Policies (CLI) on page 700

### Example: Configuring Redirect Web Filtering Custom Objects (CLI)



**NOTE:** Rather than or in addition to custom object lists, you can use included default lists and included whitelist and blacklist categories.

To configure redirect Web filtering using the CLI, you must first create your custom objects:

1. Configure a URL pattern list custom object and add values to it. The following example creates the `urllist4` custom object:

```
user@host# set security utm custom-objects url-pattern urllist4 value
[http://www.juniper.net 1.2.3.4]
```

When entering the URL pattern, note the following wildcard character support:

- The `\*\.[]\?` wildcard characters are supported.
- You must precede all wildcard URLs with `http://`.
- You can only use the asterisk `*` wildcard character if it is at the beginning of the URL and is followed by a period.
- You can only use the question mark `?` wildcard character at the end of the URL.
- The following wildcard syntax IS supported: `http://*.juniper.net`, `http://www.juniper.ne?`, `http://www.juniper.n??`.
- The following wildcard syntax is NOT supported: `*juniper.net`, `www.juniper.ne?`, `http://*juniper.net`, `http://*`.



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure custom URL category lists.

2. Configure a custom URL category list custom object by using the URL pattern list you created. The following example adds the `urllist3` URL pattern list to the `custurl4` custom object:

```
user@host# set security utm custom-objects custom-url-category custurl4 value
urllist3
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Configuring Redirect Web Filtering Feature Profiles (CLI)

After you create your custom objects, configure the Websense-redirect Web filtering feature profile.

To configure redirect Web filtering feature profiles:

1. If you are using included global whitelist and blacklist categories, select those global categories. This is the first filtering category that both integrated and redirect Web filtering use. If no match is made, the URL is forwarded to the SurfControl server. The following example selects the `Drugs_Alcohol_Tobacco` blacklist and the `Computing_Internet` whitelist:

```
user@host# set security utm feature-profile web-filtering url-blacklist
Drugs_Alcohol_Tobacco
user@host# set security utm feature-profile web-filtering url-whitelist
Computing_Internet
```

**NOTE:**

2. Select Websense-redirect as your Web filtering engine:

```
user@host# set security utm feature-profile web-filtering websense-redirect
```

3. Create a Websense-redirect profile. The following example creates the websenseprofile1 profile:

```
user@host# set security utm feature-profile web-filtering websense-redirect profile websenseprofile1
```

4. Set Websense server parameters by entering the server name or IP address. The following example specifies the Websenseserver server name:

```
user@host# set security utm feature-profile web-filtering websense-redirect profile websenseprofile1 server host Websenseserver
```

5. Enter the port number for communicating with the Websense server. (Default ports are 80, 8080, and 8081.) The following example sets the port to 8080:

```
user@host# set security utm feature-profile web-filtering websense-redirect profile websenseprofile1 server port 8080
```

6. Select fallback settings (block or log and permit) for this profile. The fallback actions are taken when errors in each category configured occur. The following example sets fallback options to block:

```
user@host# set security utm feature-profile web-filtering websense-redirect profile websenseprofile1 fallback-settings default block
user@host# set security utm feature-profile web-filtering websense-redirect profile websenseprofile1 fallback-settings server-connectivity block
user@host# set security utm feature-profile web-filtering websense-redirect profile websenseprofile1 fallback-settings timeout block
user@host# set security utm feature-profile web-filtering websense-redirect profile websenseprofile1 fallback-settings too-many-requests block
```

7. Enter the number of sockets used for communicating between the client and server. The default is 1. The following example sets the sockets to 1:

```
user@host# set security utm feature-profile web-filtering websense-redirect profile websenseprofile1 sockets 1
```

8. Enter a timeout limit for requests. Once this limit is reached, fail mode settings are applied. The default here is 10 seconds. You can enter a value from 10 to 240 seconds.

```
user@host# set security utm feature-profile web-filtering websense-redirect profile websenseprofile1 timeout 10
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Configuring Redirect Web Filtering UTM Policies (CLI)

To configure a UTM policy:

1. Create the UTM policy. The following example creates the utmp6 policy:

```
user@host# set security utm utm-policy utmp6
```

2. Attach the UTM policy to a profile. The following example attaches the utmp6 policy to the websenseprofile1 profile:

```
user@host# set security utm utm-policy utmp6 web-filtering http-profile websenseprofile1
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

#### Example: Attaching Redirect Web Filtering UTM Policies to Security Policies (CLI)

Attach the UTM policy to the security policy. The following example attaches the utmp6 UTM policy to the p6 security policy:

```
user@host# set security policies from-zone trust to-zone untrust policy p6 then permit application-services utm-policy utmp6
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

---

## Local Web Filtering

- Understanding Local Web Filtering on page 700
- Example: Configuring Local Web Filtering (CLI) on page 702

### Understanding Local Web Filtering

With local Web filtering, the firewall intercepts every HTTP request in a TCP connection and extracts the URL. The decision making is done on the device after it looks up a URL to determine if it is in the whitelist or blacklist based on its user-defined category. If the URL is in the url-blacklist, the request is blocked; if it's in the url-whitelist, the request is permitted. If the URL is not in either list, the defined default action will occur (block, log-and-permit, or permit). You can permit or block access to a requested site by binding a Web filtering profile to a firewall policy. Local Web filtering provides basic Web filtering without requiring an additional license or external category server.

This topic contains the following sections:

- User-Defined URL Categories on page 700
- Local Web Filtering Process on page 701
- Local Web Filtering Profiles on page 701
- Profile Matching Precedence on page 701

### User-Defined URL Categories

When defining your own URL categories, you can group URLs and create categories specific to your needs. Each category can have a maximum of 20 URLs. When you create a category, you can add either the URL or the IP address of a site. When you add a URL to a user-defined category, the device performs DNS lookup, resolves the hostname into IP addresses, and caches this information. When a user tries to access a site with the IP address of the site, the device checks the cached list of IP addresses and tries to resolve the hostname. Many sites have dynamic IP addresses, meaning that their IP addresses change periodically. A user attempting to access a site can type an IP address that is not

in the cached list on the device. Therefore, if you know the IP addresses of sites you are adding to a category, enter both the URL and the IP address(es) of the site.

You define your own categories using URL pattern list and custom URL category list custom objects. Once defined, you assign your categories to the global user-defined url-blacklist (block) or url-whitelist (permit) categories.



**NOTE:** Web filtering is performed on all the methods defined in HTTP 1.0 and HTTP 1.1.

## Local Web Filtering Process

This is a general description of how Web traffic is intercepted and acted upon by the Web filtering module.

1. The device intercepts a TCP connection.
2. The device intercepts each HTTP request in the TCP connection.
3. The device extracts each URL in the HTTP request and checks its URL against the user-defined whitelist and blacklist.
4. If the URL is found in the blacklist, the request is not permitted and a deny page is sent to the http client. If the URL is found in the whitelist, the request is permitted.
5. If the URL is not found in the whitelist or blacklist, the configured default fallback action is applied. If no fallback action is defined, then the request is permitted.

## Local Web Filtering Profiles

You configure Web filtering profiles that permit or block URLs according to defined custom categories. A Web filtering profile consists of a group of URL categories assigned one of the following actions:

- Blacklist — The device always blocks access to the websites in this list. Only user-defined categories are used with local Web filtering.
- Whitelist — The device always allows access to the websites in this list. Only user-defined categories are used with local Web filtering.

A Web filtering profile can contain one blacklist or one whitelist with multiple user-defined categories each with a permit or block action. You can define a default fallback action when the incoming URL does not belong to any of the categories defined in the profile. If the action for the default category is block, the incoming URL is blocked if it does not match any of the categories explicitly defined in the profile. If an action for the default action is not specified, the default action of permit is applied to the incoming URL not matching any category.

## Profile Matching Precedence

When a profile employs several categories for URL matching, those categories are checked for matches in the following order:

1. If present, the global blacklist is checked first. If a match is made, the URL is blocked. If no match is found...
2. The global whitelist is checked next. If a match is made, the URL is permitted. If no match is found...
3. User-defined categories are checked next. If a match is made, the URL is blocked or permitted as specified.

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Configuring Local Web Filtering (CLI)

- Example: Configuring Local Web Filtering Custom Objects (CLI) on page 702
- Example: Configuring Local Web Filtering Feature Profiles (CLI) on page 703
- Example: Configuring Local Web Filtering UTM Policies (CLI) on page 704
- Example: Attaching Local Web Filtering UTM Policies to Security Policies (CLI) on page 704

#### Example: Configuring Local Web Filtering Custom Objects (CLI)

To create your custom objects:

1. Configure a URL pattern list custom object by creating the list name and adding values to it. The following examples create the `urllist3` list and the `urllist4` list:

```
user@host# set security utm custom-objects url-pattern urllist3 value
[http://www.juniper.net 1.2.3.4]
user@host# set security utm custom-objects url-pattern urllist4 value
[http://www.acmegizmo.com 1.2.3.4]
```

When entering the URL pattern, note the following wildcard character support:

- The `\*\.[\]\?*` wildcard characters are supported.
- You must precede all wildcard URLs with `http://`.
- You can only use the asterisk `*` wildcard character if it is at the beginning of the URL and is followed by a period.
- You can only use the question mark `?` wildcard character at the end of the URL.
- The following wildcard syntax IS supported: `http://*.juniper.net`, `http://www.juniper.ne?`, `http://www.juniper.n??`.
- The following wildcard syntax is NOT supported: `*juniper.net`, `www.juniper.ne?`, `http://*juniper.net`, `http://*`.



**NOTE:** Because you use URL pattern lists to create custom URL category lists, you must configure URL pattern list custom objects before you configure custom URL category lists. The URL and IP address is added in this example.

---

2. Configure a custom URL category list custom object by using the URL pattern list you created. The following examples add the custurl3 and custurl4 category lists to the urllist3 and urllist4 URL pattern lists, respectively:

```
user@host# set security utm custom-objects custom-url-category custurl3 value
urllist3
user@host# set security utm custom-objects custom-url-category custurl4 value
urllist4
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Configuring Local Web Filtering Feature Profiles (CLI)

After you create custom objects, configure the juniper-local Web filtering feature profile:

1. If you are using included global whitelist and blacklist categories, select those global categories. This is the first filtering category that both integrated, redirect, and local Web filtering use. If no match is made, the configured default fallback action is performed. The following example creates the custurl3 blacklist and the custurl4 whitelist:

```
user@host# set security utm feature-profile web-filtering url-blacklist custurl3
user@host# set security utm feature-profile web-filtering url-whitelist custurl4
```



**NOTE:** In this example, the user-defined category is assigned to the global url-whitelist category. This will permit all URLs in that category.

2. Select juniper-local as your Web filtering engine:

```
user@host# set security utm feature-profile web-filtering type juniper-local
```

3. Create a juniper-local profile by creating a profile with a default action (permit, log and permit, block) for requests that experience errors. The following example creates the localprofile1 profile with a default action of permit:

```
user@host# set security utm feature-profile web-filtering juniper-local profile
localprofile1 default permit
```

4. Enter a custom message to be sent when HTTP requests are blocked. The following example creates a custom message that says "Access to this site is not permitted":

```
user@host# set security utm feature-profile web-filtering juniper-local profile
localprofile1 custom-block-message "Access to this site is not permitted"
```

5. Select fallback settings (block or log and permit) for this profile. The fallback actions are taken when errors in each configured category occur. The following example sets fallback options to block:

```
user@host# set security utm feature-profile web-filtering juniper-local profile
localprofile1 fallback-settings default block
user@host# set security utm feature-profile web-filtering juniper-local profile
localprofile1 fallback-settings too-many-requests block
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Configuring Local Web Filtering UTM Policies (CLI)

To configure a UTM policy:

1. Create the UTM policy. The following example creates the utmp5 policy:  

```
user@host# set security utm utm-policy utmp5
```
2. Attach the UTM policy to a profile. The following example attaches the utmp6 policy to the localprofile1 profile:  

```
user@host# set security utm utm-policy utmp5 web-filtering http-profile localprofile1
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Attaching Local Web Filtering UTM Policies to Security Policies (CLI)

To attach a UTM policy to a security policy:

1. Create and configure the security policy. The following example creates the p5 security policy:  

```
user@host# set security policies from-zone trust to-zone untrust policy p5 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy p5 match destination-address any
user@host# set security policies from-zone trust to-zone untrust policy p5 match application junos-http
```
2. Attach the UTM policy to the security policy. The following example attaches the utmp5 UTM policy to the p5 security policy:  

```
user@host# set security policies from-zone trust to-zone untrust policy p5 then permit application-services utm-policy utmp5
```

**Related Topics** *Junos OS Feature Support Reference for SRX Series and J Series Devices*

---

## Monitoring Web Filtering Configurations

**Purpose** View Web filtering statistics.

**Action** To view Web filtering statistics using the CLI, enter the following commands:

```
user@host > show security utm web-filtering status
user@host > show security utm web-filtering statistics
```

To view Web filtering statistics using J-Web:

1. Select **Monitor>UTM>Web Filtering**.

The following information becomes viewable in the right pane.

```
white list hit: #
Black list hit: #
Queries to server: #
Server reply permit: #
Server reply block: #
```



Custom category permit: #  
Custom category block: #  
Cache hit permit: #  
Cache hit block: #  
Web-filtering sessions in total: #  
Web-filtering sessions in use: #  
Fall back: log-and-permit block  
Default # #  
Timeout # #  
Connectivity # #  
Too-many-requests # #

2. You can click the **Clear Web Filtering STAT** button to clear all current viewable statistics and begin collecting new statistics.

**Related Topics**    *Junos OS Feature Support Reference for SRX Series and J Series Devices*



## PART 9

# Attack Detection and Prevention

- Attack Detection and Prevention on page 709
- Reconnaissance Deterrence on page 711
- Suspicious Packet Attributes on page 735
- Denial-of-Service Attacks on page 743



# Attack Detection and Prevention

- Attack Detection and Prevention Overview on page 709

## Attack Detection and Prevention Overview

---

The Juniper Networks Intrusion Detection and Prevention (IDP) feature, also known as a *stateful firewall*, detects and prevents attacks in network traffic.

An exploit can be either an information-gathering probe or an attack to compromise, disable, or harm a network or network resource. In some cases, the distinction between the two objectives of an exploit can be unclear. For example, a barrage of TCP SYN segments might be an IP address sweep with the intent of triggering responses from active hosts, or it might be a SYN flood attack with the intent of overwhelming a network so that it can no longer function properly. Furthermore, because an attacker usually precedes an attack by performing reconnaissance on the target, we can consider information-gathering efforts as a precursor to an impending attack—that is, they constitute the first stage of an attack. Thus, the term *exploit* encompasses both reconnaissance and attack activities, and the distinction between the two is not always clear.

Juniper Networks provides various detection and defense mechanisms at the zone and policy levels to combat exploits at all stages of their execution:

- Screen options at the zone level.
- Firewall policies at the inter-, intra-, and super-zone policy levels (*super-zone* here means in global policies, where no security zones are referenced).

To secure all connection attempts, Junos OS uses a dynamic packet-filtering method known as *stateful inspection*. Using this method, Junos OS identifies various components in the IP packet and TCP segment headers—source and destination IP addresses, source and destination port numbers, and packet sequence numbers—and maintains the state of each TCP session and pseudo UDP session traversing the firewall. (Junos OS also modifies session states based on changing elements such as dynamic port changes or session termination.) When a responding TCP packet arrives, Junos OS compares the information reported in its header with the state of its associated session stored in the inspection table. If they match, the responding packet is allowed to pass the firewall. If the two do not match, the packet is dropped.

Junos OS screen options secure a zone by inspecting, then allowing or denying, all connection attempts that require crossing an interface bound to that zone. Junos OS then applies firewall policies, which can contain content filtering and IDP components, to the traffic that passes the screen filters.

**Related Topics**   • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## CHAPTER 36

# Reconnaissance Deterrence

- Reconnaissance Deterrence Overview on page 711
- IP Address Sweeps on page 711
- Port Scanning on page 713
- Network Reconnaissance Using IP Options on page 715
- Operating System Probes on page 720
- Attacker Evasion Techniques on page 725

## Reconnaissance Deterrence Overview

---

Attackers can better plan their attack when they first know the layout of the targeted network (which IP addresses have active hosts), the possible entry points (which port numbers are active on the active hosts), and the constitution of their victims (which operating system the active hosts are running). To gain this information, attackers must perform reconnaissance.

Juniper Networks provides several screen options for deterring attackers' reconnaissance efforts and thereby hindering them from obtaining valuable information about the protected network and network resources.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## IP Address Sweeps

---

- Understanding IP Address Sweeps on page 711
- Example: Blocking IP Address Sweeps on page 712

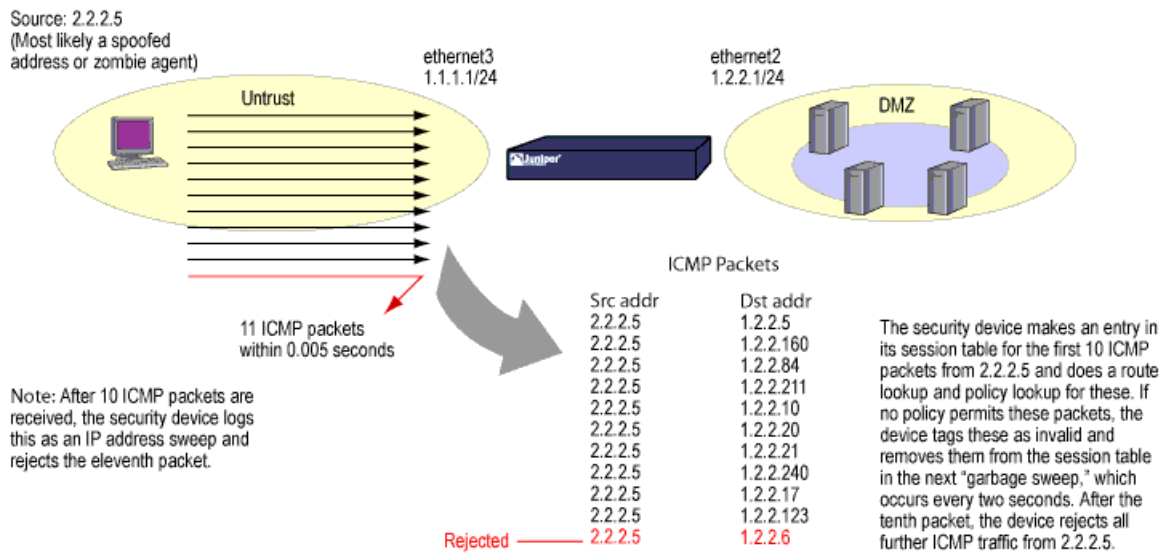
## Understanding IP Address Sweeps

An address sweep occurs when one source IP address sends a defined number of ICMP packets sent to different hosts within a defined interval (5000 microseconds is the default). The purpose of this attack is to send ICMP packets—typically echo requests—to various hosts in the hopes that at least one replies, thus uncovering an address to target.

Junos OS internally logs the number of ICMP packets to different addresses from one remote source. Using the default settings, if a remote host sends ICMP traffic to 10 addresses in 0.005 seconds (5000 microseconds), then the device flags this as an

address sweep attack and rejects all further ICMP packets from that host for the remainder of the specified threshold time period. See Figure 57 on page 712.

Figure 57: Address Sweep



Consider enabling this screen option for a security zone only if there is a policy permitting ICMP traffic from that zone. Otherwise, you do not need to enable the screen option. The lack of such a policy denies all ICMP traffic from that zone, precluding an attacker from successfully performing an IP address sweep anyway.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Example: Blocking IP Address Sweeps

This example describes how to configure a screen to block an IP address sweep originating from a security zone.

- Requirements on page 712
- Overview on page 712
- Configuration on page 713
- Verification on page 713

### Requirements

Before you begin:

1. Understand how IP address sweeps work. See "Understanding IP Address Sweeps" on page 711.
2. Configure security zones. "Security Zones and Interfaces Overview" on page 85.

### Overview

You need to enable a screen for a security zone if you have configured a policy that permits ICMP traffic from that zone. If you have not configured such a policy, then your system



denies all ICMP traffic from that zone, and the attacker cannot perform an IP address sweep successfully anyway.

In this example you configure a **5000-ip-sweep** screen to block IP address sweeps originating in the zone-1 security zone.

### Configuration

#### Step-by-Step Procedure

To configure a screen to block IP address sweeps:

1. Configure a screen.

```
[edit]
user@host# set security screen ids-option 5000-ip-sweep icmp ip-sweep threshold
5000
```

2. Enable the screen in the security zone.

```
[edit]
user@host# set security zones security-zone zone-1 screen 5000-ip-sweep
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security zones** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Port Scanning

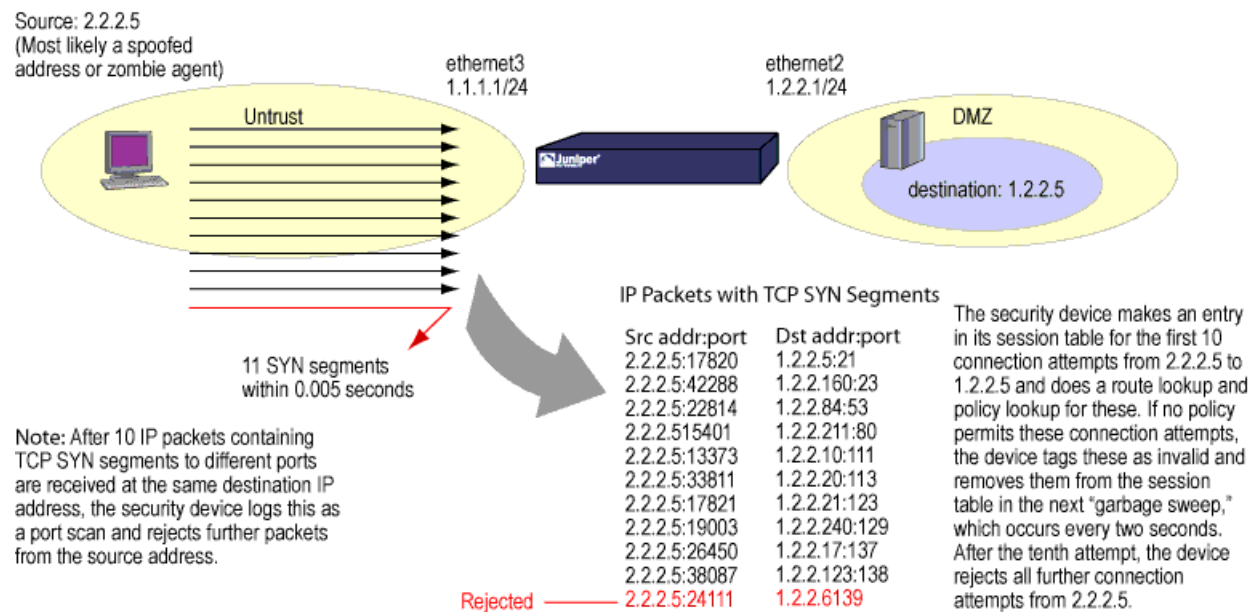
- Understanding Port Scanning on page 713
- Example: Blocking Port Scans on page 714

### Understanding Port Scanning

A port scan occurs when one source IP address sends IP packets containing TCP SYN segments to a defined number of different ports at the same destination IP address within a defined interval (5000 microseconds is the default). The purpose of this attack is to scan the available services in the hopes that at least one port will respond, thus identifying a service to target.

Junos OS internally logs the number of different ports scanned from one remote source. Using the default settings, if a remote host scans 10 ports in 0.005 seconds (5000 microseconds), then the device flags this as a port scan attack and rejects all further packets from the remote source, regardless of the destination IP address, for the remainder of the specified timeout period. See Figure 58 on page 714.

Figure 58: Port Scan



**Related Topics** • [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)

## Example: Blocking Port Scans

This example shows how to configure a screen to block port scans originating from a particular security zone.

- Requirements on page 714
- Overview on page 714
- Configuration on page 714
- Verification on page 715

### Requirements

Before you begin, understand how port scanning works. See "Understanding Port Scanning" on page 713.

### Overview

You can use a port scan to block IP packets containing TCP SYN segments sent to different ports from the same destination address within a defined interval.

In this example, you configure a 5000-port-scan screen to block port scans originating from a particular security zone.

### Configuration

**Step-by-Step Procedure** To configure a screen to block port scans:

1. Configure the screen.

```
[edit]
user@host# set security screen ids-option 5000-port-scan tcp port-scan threshold
5000
```

- If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security screen** command.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Network Reconnaissance Using IP Options

- Understanding Network Reconnaissance Using IP Options on page 715
- Example: Detecting Packets That Use IP Screen Options for Reconnaissance on page 718

### Understanding Network Reconnaissance Using IP Options

The IP standard RFC 791, *Internet Protocol*, specifies a set of options for providing special routing controls, diagnostic tools, and security.

RFC 791 states that these options are “unnecessary for the most common communications” and, in reality, they rarely appear in IP packet headers. These options appear after the destination address in an IP packet header, as shown in Figure 59 on page 715. When they do appear, they are frequently being put to some illegitimate use.

**Figure 59: Routing Options**

|                     |        |                 |                                |   |   |
|---------------------|--------|-----------------|--------------------------------|---|---|
| Version             | Header | Type of Service | Total Packet Length (in Bytes) |   |   |
| Identification      |        |                 | 0                              | D | M |
| Fragment Offset     |        |                 | Header Checksum                |   |   |
| Time to Live (TTL)  |        | Protocol        | Header Checksum                |   |   |
| Source Address      |        |                 |                                |   |   |
| Destination Address |        |                 |                                |   |   |
| Options             |        |                 |                                |   |   |
| Payload             |        |                 |                                |   |   |

This topic contains the following sections:

- Uses for IP Packet Header Options on page 715
- Screen Options for Detecting IP Options Used for Reconnaissance on page 717

### Uses for IP Packet Header Options

Table 69 on page 716 lists the IP options and their accompanying attributes.

Table 69: IP Options and Attributes

| Type               | Class | Number | Length  | Intended Use                                                                                                                                                                                                                                                                                                                                                                    | Nefarious Use                                                                                                                                                                                                        |
|--------------------|-------|--------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| End of Options     | 0*    | 0      | 0       | Indicates the end of one or more IP options.                                                                                                                                                                                                                                                                                                                                    | None.                                                                                                                                                                                                                |
| No Options         | 0     | 1      | 0       | Indicates there are no IP options in the header.                                                                                                                                                                                                                                                                                                                                | None.                                                                                                                                                                                                                |
| Security           | 0     | 2      | 11 bits | <p>Provides a way for hosts to send security, TCC (closed user group) parameters, and Handling Restriction Codes compatible with Department of Defense (DoD) requirements. (This option, as specified in RFC 791, <i>Internet Protocol</i>, and RFC 1038, <i>Revised IP Security Option</i>, is obsolete.)</p> <p>Currently, this screen option is applicable only to IPv4.</p> | Unknown. However, because it is obsolete, its presence in an IP header is suspect.                                                                                                                                   |
| Loose Source Route | 0     | 3      | Varies  | Specifies a partial route list for a packet to take on its journey from source to destination. The packet must proceed in the order of addresses specified, but it is allowed to pass through other devices in between those specified.                                                                                                                                         | Evasion. The attacker can use the specified routes to hide the true source of a packet or to gain access to a protected network.                                                                                     |
| Record Route       | 0     | 7      | Varies  | <p>Records the IP addresses of the network devices along the path that the IP packet travels. The destination machine can then extract and process the route information. (Due to the size limitation of 40 bytes for both the option and storage space, this can only record up to 9 IP addresses.)</p> <p>Currently, this screen option is applicable only to IPv4.</p>       | Reconnaissance. If the destination host is a compromised machine in the attacker's control, he or she can glean information about the topology and addressing scheme of the network through which the packet passed. |
| Stream ID          | 0     | 8      | 4 bits  | <p>(Obsolete) Provided a way for the 16-bit SATNET stream identifier to be carried through networks that did not support the stream concept.</p> <p>Currently, this screen option is applicable only to IPv4.</p>                                                                                                                                                               | Unknown. However, because it is obsolete, its presence in an IP header is suspect.                                                                                                                                   |

Table 69: IP Options and Attributes (*continued*)

| Type                | Class | Number | Length | Intended Use                                                                                                                                                                                                                                                                                                                                                                                                                             | Nefarious Use                                                                                                                                                                                                            |
|---------------------|-------|--------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Strict Source Route | 0     | 9      | Varies | Specifies the complete route list for a packet to take on its journey from source to destination. The last address in the list replaces the address in the destination field.<br><br>Currently, this screen option is applicable only to IPv4.                                                                                                                                                                                           | Evasion. An attacker can use the specified routes to hide the true source of a packet or to gain access to a protected network.                                                                                          |
| Timestamp           | 2**   | 4      |        | Records the time (in coordinated universal time [UTC]***) when each network device receives the packet during its trip from the point of origin to its destination. The network devices are identified by IP address.<br><br>This option develops a list of IP addresses of the devices along the path of the packet and the duration of transmission between each one.<br><br>Currently, this screen option is applicable only to IPv4. | Reconnaissance. If the destination host is a compromised machine in the attacker's control, he or she can glean information about the topology and addressing scheme of the network through which the packet has passed. |

\* The class of options identified as 0 was designed to provide extra packet or network control.

\*\* The class of options identified as 2 was designed for diagnostics, debugging, and measurement.

\*\*\* The timestamp uses the number of milliseconds since midnight UTC. UTC is also known as Greenwich Mean Time (GMT), which is the basis for the international time standard.

## Screen Options for Detecting IP Options Used for Reconnaissance

The following screen options detect IP options that an attacker can use for reconnaissance or for some unknown but suspect purpose:

- Record Route—Junos OS detects packets where the IP option is 7 (record route) and records the event in the screen counters list for the ingress interface. Currently, this screen option is applicable only to IPv4.
- Timestamp—Junos OS detects packets where the IP option list includes option 4 (Internet timestamp) and records the event in the screen counters list for the ingress interface. Currently, this screen option is applicable only to IPv4.
- Security—Junos OS detects packets where the IP option is 2 (security) and records the event in the screen counters list for the ingress interface. Currently, this screen option is applicable only to IPv4.
- Stream ID—Junos OS detects packets where the IP option is 8 (stream ID) and records the event in the screen counters list for the ingress interface. Currently, this screen option is applicable only to IPv4.

If a packet with any of the previous IP options is received, Junos OS flags this as a network reconnaissance attack and records the event for the ingress interface.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Example: Detecting Packets That Use IP Screen Options for Reconnaissance

This example shows how to detect packets that use IP screen options for reconnaissance.

- Requirements on page 718
- Overview on page 718
- Configuration on page 718
- Verification on page 719

### Requirements

Before you begin, understand how network reconnaissance works. See “Understanding Network Reconnaissance Using IP Options” on page 715.

### Overview

RFC 791, *Internet Protocol*, specifies a set of options for providing special routing controls, diagnostic tools, and security. The screen options detect IP options that an attacker can use for reconnaissance, including record route, timestamp, security, and stream ID.

In this example, you configure the following IP screens: ip-record-route, ip-timestamp-opt, ip-security-opt, and ip-stream-opt. The screens are enabled in the zone-1 security zone.

### Configuration

#### CLI Quick Configuration

To quickly detect packets with the record route, timestamp, security, and stream ID IP screen options, copy the following commands and paste them into the CLI.

```
[edit]
set security screen ids-option ip-record-route ip record-route-option
set security screen ids-option ip-timestamp-opt ip timestamp-option
set security screen ids-option ip-security-opt ip security-option
set security screen ids-option ip-stream-opt ip stream-option
set security zones security-zone zone-1 screen ip-record-route-opt
set security zones security-zone zone-1 screen ip-timestamp-opt
set security zones security-zone zone-1 screen ip-security-opt
set security zones security-zone zone-1 screen ip-stream-opt
```

#### Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To detect packets that use IP screen options for reconnaissance:

1. Configure IP screen options.



NOTE: Currently, these screen options support IPv4 only.

```
[edit security screen]
user@host# set ids-option ip-record-route ip record-route-option
user@host# set ids-option ip-timestamp-opt ip timestamp-option
user@host# set ids-option ip-security-opt ip security-option
user@host# set ids-option ip-stream-opt ip stream-option
```

2. Enable the screens in the security zone:

```
[edit security zones]
user@host# set zone screen ip-record-route-opt
user@host# set security-zone zone-1 screen ip-timestamp-opt
user@host# set security-zone zone-1 screen ip-security-opt
user@host# set security-zone zone-1 screen ip-stream-opt
```

**Results** From configuration mode, confirm your configuration by entering the **show security screen** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security screen
ids-option ip-record-route {
 ip {
 record-route-option;
 }
}
ids-option ip-security-opt {
 ip {
 security-option;
 }
}
ids-option ip-stream-opt {
 ip {
 stream-option;
 }
}
ids-option ip-timestamp-opt {
 ip {
 timestamp-option;
 }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying the Detection Packets That Use IP Options for Reconnaissance on page 720
- Verifying Screens in the Zone-1 Security Zone Are Enabled on page 720

**Verifying the Detection Packets That Use IP Options for Reconnaissance**

**Purpose** Verify that the IP screen options for reconnaissance are configured.

**Action** From operational mode, enter the **show security screen** command.

**Verifying Screens in the Zone-1 Security Zone Are Enabled**

**Purpose** Verify that the screens in the zone-1 security zone are enabled.

**Action** From operational mode, enter the **show security zones** command.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

---

## Operating System Probes

- Understanding Operating System Probes on page 720
- TCP Headers with SYN and FIN Flags Set on page 720
- TCP Headers With FIN Flag Set and Without ACK Flag Set on page 722
- TCP Header with No Flags Set on page 724

### Understanding Operating System Probes

Before launching an exploit, attackers might try to probe the targeted host to learn its operating system (OS). With that knowledge, they can better decide which attack to launch and which vulnerabilities to exploit. Junos OS can block reconnaissance probes commonly used to gather information about OS types.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### TCP Headers with SYN and FIN Flags Set

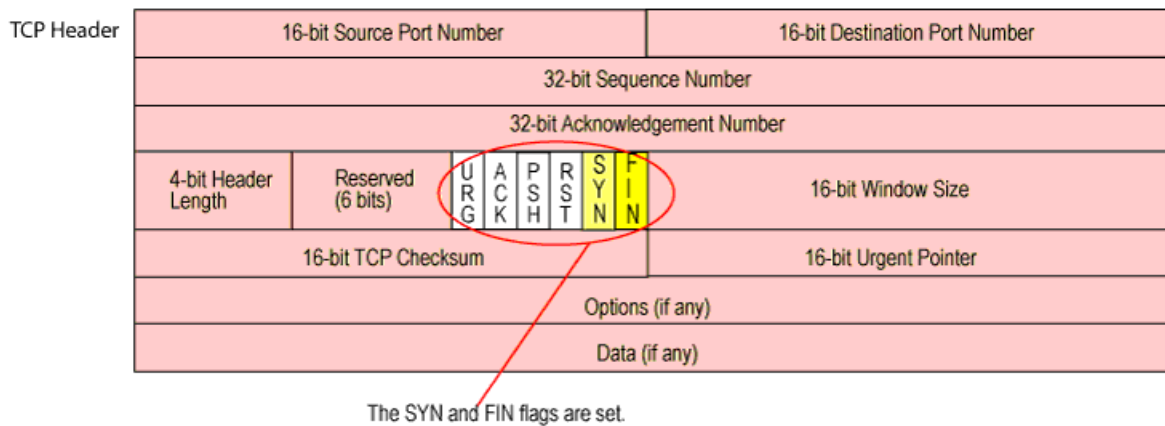
- Understanding TCP Headers with SYN and FIN Flags Set on page 720
- Example: Blocking Packets with SYN and FIN Flags Set on page 721

#### Understanding TCP Headers with SYN and FIN Flags Set

Both the SYN and FIN control flags are not normally set in the same TCP segment header. The SYN flag synchronizes sequence numbers to initiate a TCP connection. The FIN flag indicates the end of data transmission to finish a TCP connection. Their purposes are mutually exclusive. A TCP header with the SYN and FIN flags set is anomalous TCP behavior, causing various responses from the recipient, depending on the OS. See Figure 60 on page 721.



Figure 60: TCP Header with SYN and FIN Flags Set



An attacker can send a segment with both flags set to see what kind of system reply is returned and thereby determine what kind of OS is on the receiving end. The attacker can then use any known system vulnerabilities for further attacks.

When you enable this screen option, Junos OS checks if the SYN and FIN flags are set in TCP headers. If it discovers such a header, it drops the packet.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Blocking Packets with SYN and FIN Flags Set

This example shows how to create a screen to block packets with the SYN and FIN flags set.

- Requirements on page 721
- Overview on page 721
- Configuration on page 721
- Verification on page 722

#### Requirements

Before you begin, understand how TCP headers with SYN and FIN flags work. See “Understanding TCP Headers with SYN and FIN Flags Set” on page 720.

#### Overview

The TCP header with the SYN and FIN flags set cause different responses from a targeted device depending on the OS it is running. The syn-fin screen is enabled for the security zone.

In this example, you create a screen called screen-1 in a security zone to block packets with the SYN and FIN flags set.

#### Configuration

**Step-by-Step Procedure** To block packets with both the SYN and FIN flags set:

1. Configure the screen.

```
[edit]
user@host# set security screen ids-option screen-1 tcp syn-fin
```

2. Enable the screen in the security zone.

```
[edit]
user@host# set security zones security-zone zone-1 screen screen-1
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

#### Verification

To verify the configuration is working properly, enter the **show security** command.

### TCP Headers With FIN Flag Set and Without ACK Flag Set

- Understanding TCP Headers With FIN Flag Set and Without ACK Flag Set on page 722
- Example: Blocking Packets With FIN Flag Set and Without ACK Flag Set on page 723

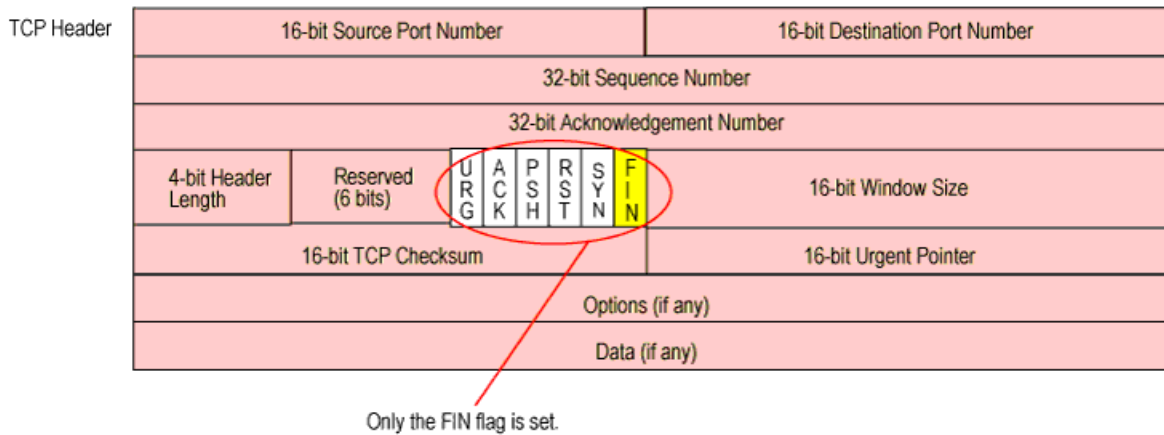
#### Understanding TCP Headers With FIN Flag Set and Without ACK Flag Set

Figure 61 on page 723 shows TCP segments with the FIN control flag set (to signal the conclusion of a session and terminate the connection). Normally, TCP segments with the FIN flag set also have the ACK flag set (to acknowledge the previous packet received). Because a TCP header with the FIN flag set but not the ACK flag is anomalous TCP behavior, there is no uniform response to this. The OS might respond by sending a TCP segment with the RST flag set. Another might completely ignore it. The victim's response can provide the attacker with a clue as to its OS. (Other purposes for sending a TCP segment with the FIN flag set are to evade detection while performing address and port scans and to evade defenses on guard for a SYN flood by performing a FIN flood instead.)



NOTE: Vendors have interpreted RFC 793, *Transmission Control Protocol*, variously when designing their TCP/IP implementations. When a TCP segment arrives with the FIN flag set but not the ACK flag, some implementations send RST segments, while others drop the packet without sending an RST.

Figure 61: TCP Header with FIN Flag Set



When you enable this screen option, Junos OS checks if the FIN flag is set but not the ACK flag in TCP headers. If it discovers a packet with such a header, it drops the packet.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

#### Example: Blocking Packets With FIN Flag Set and Without ACK Flag Set

This example shows how to create a screen to block packets with the FIN flag set but the ACK flag not set.

- Requirements on page 723
- Overview on page 723
- Configuration on page 723
- Verification on page 724

#### Requirements

Before you begin, understand how TCP headers work. See "Understanding TCP Headers With FIN Flag Set and Without ACK Flag Set" on page 722.

#### Overview

The TCP segments with the FIN flag set also have the ACK flag set to acknowledge the previous packet received. Because a TCP header with the FIN flag set but the ACK flag not set is anomalous TCP behavior, there is no uniform response to this. When you enable the fin-no-ack screen option, Junos OS checks if the FIN flag is set but not the ACK flag in TCP headers. If it discovers a packet with such a header, it drops the packet.

In this example, you create a screen called screen-1 to block packets with the FIN flag set but the ACK flag not set.

#### Configuration

**Step-by-Step Procedure** To block packets with the FIN flag set but the ACK flag not set:

1. Configure the screen.

[edit]

```
user@host# set security screen ids-option screen-1 tcp fin-no-ack
```

- If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security screen** command.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

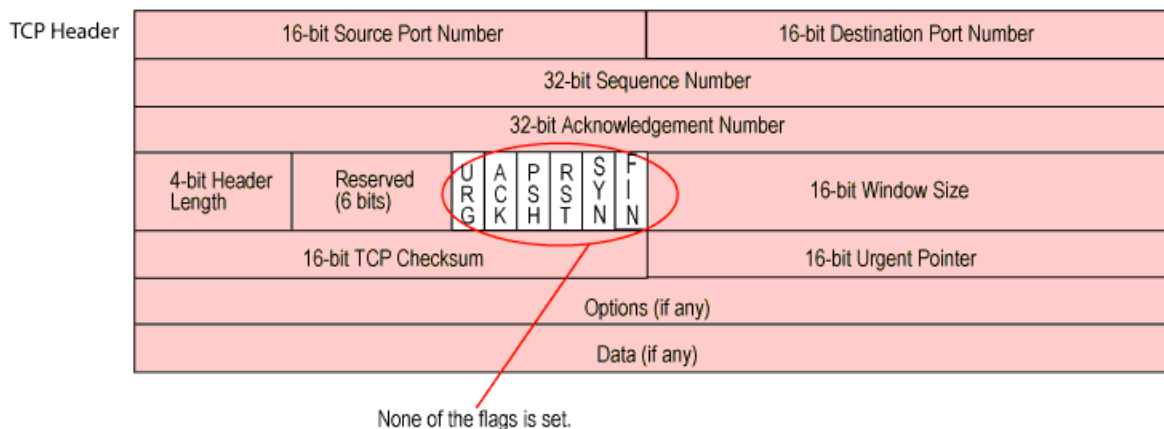
## TCP Header with No Flags Set

- Understanding TCP Header with No Flags Set on page 724
- Example: Blocking Packets with No Flags Set on page 724

### Understanding TCP Header with No Flags Set

A normal TCP segment header has at least one flag control set. A TCP segment with no control flags set is an anomalous event. Because different operating systems respond differently to such anomalies, the response (or lack of response) from the targeted device can provide a clue as to the type of OS it is running. See Figure 62 on page 724.

**Figure 62: TCP Header with No Flags Set**



When you enable the device to detect TCP segment headers with no flags set, the device drops all TCP packets with a missing or malformed flags field.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Blocking Packets with No Flags Set

This example shows how to create a screen to block packets with no flags set.

- Requirements on page 725
- Overview on page 725

- Configuration on page 725
- Verification on page 725

### Requirements

Before you begin, understand how a TCP header with no flags set works. See “Understanding TCP Header with No Flags Set” on page 724.

### Overview

A normal TCP segment header has at least one flag control set. A TCP segment with no control flags set is an anomalous event. Because different operating systems respond differently to such anomalies, the response (or lack of response) from the targeted device can provide a clue as to the type of OS it is running.

When you enable the device to detect TCP segment headers with no flags set, the device drops all TCP packets with a missing or malformed flags field.

In this example, you create a screen called screen-1 to block packets with no flags set.

### Configuration

#### Step-by-Step Procedure

To block packets with no flags set:

1. Configure the screen.  

```
[edit]
user@host# set security screen ids-option screen-1 tcp tcp-no-flag
```
2. Enable the screen in the security zone.  

```
[edit]
user@host# set security zones security-zone zone-1 screen screen-1
```
3. If you are done configuring the device, commit the configuration.  

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security screen** command.

## Attacker Evasion Techniques

- Understanding Attacker Evasion Techniques on page 725
- Fin Scanning on page 726
- TCP SYN Checking on page 726
- IP Spoofing on page 729
- IP Source Route Options on page 730

## Understanding Attacker Evasion Techniques

Whether gathering information or launching an attack, it is generally expected that the attacker avoids detection. Although some IP address and port scans are blatant and

easily detectable, more wily attackers use a variety of means to conceal their activity. Techniques such as using FIN scans instead of SYN scans—which attackers know most firewalls and intrusion detection programs detect—indicate an evolution of reconnaissance and exploit techniques for evading detection and successfully accomplishing their tasks.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Fin Scanning

- Understanding FIN Scans on page 726
- Thwarting a FIN Scan (CLI Procedure) on page 726

### Understanding FIN Scans

A FIN scan sends TCP segments with the FIN flag set in an attempt to provoke a response (a TCP segment with the RST flag set) and thereby discover an active host or an active port on a host. Attackers might use this approach rather than perform an address sweep with ICMP echo requests or an address scan with SYN segments, because they know that many firewalls typically guard against the latter two approaches but not necessarily against FIN segments. The use of TCP segments with the FIN flag set might evade detection and thereby help the attackers succeed in their reconnaissance efforts.

### Thwarting a FIN Scan (CLI Procedure)

To thwart FIN scans, take either or both of the following actions:

- Enable the screen option that specifically blocks TCP segments with the FIN flag set but not the ACK flag, which is anomalous for a TCP segment:

```
user@host#set security screen fin-no-ack tcp fin-no-ack
user@host#set security zones security-zone name screen fin-no-ack
```

where ***name*** is the name of the zone to which you want to apply this screen option .

- Change the packet processing behavior to reject all non-SYN packets that do not belong to an existing session. The SYN check flag is set as the default.



**NOTE:** Changing the packet flow to check that the SYN flag is set for packets that do not belong to existing sessions also thwarts other types of non-SYN scans, such as a null scan (when no TCP flags are set).

---

## TCP SYN Checking

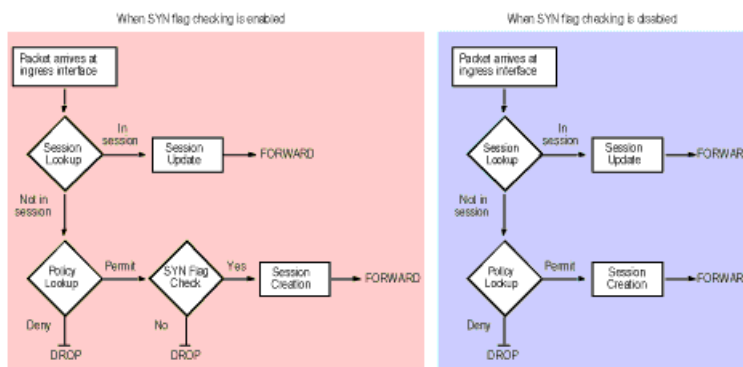
- Understanding TCP SYN Checking on page 726
- Setting TCP SYN Checking (CLI Procedure) on page 728
- Setting Strict SYN Checking (CLI Procedure) on page 729

### Understanding TCP SYN Checking

By default, Junos OS checks for SYN flags in the first packet of a session and rejects any TCP segments with non-SYN flags attempting to initiate a session. You can leave this packet flow as is or change it so that Junos OS does not enforce SYN flag checking before

creating a session. Figure 63 on page 727 illustrates packet flow sequences both when SYN flag checking is enabled and when it is disabled.

Figure 63: SYN Flag Checking



When Junos OS with SYN flag checking enabled receives a non-SYN TCP segment that does not belong to an existing session, it drops the packet. By default, Junos OS does not send a TCP RST to the source host on receiving the non-SYN segment. You can configure the device to send TCP RST to the source host by using the **set security zones security-zone trust tcp-rst** command. If the code bit of the initial non-SYN TCP packet is RST, the device does not send a TCP-RST.

Not checking for the SYN flag in the first packets offers the following advantages:

- **NSRP with Asymmetric Routing**—In an active/active NSRP configuration in a dynamic routing environment, a host might send the initial TCP segment with the SYN flag set to one device (Device-A), but the SYN/ACK might be routed to the other device in the cluster (Device-B). If this asymmetric routing occurs after Device-A has synchronized its session with Device-B, all is well. On the other hand, if the SYN/ACK response reaches Device-B before Device-A synchronizes the session and SYN checking is enabled, Device-B rejects the SYN/ACK, and the session cannot be established. With SYN checking disabled, Device-B accepts the SYN/ACK response—even though there is no existing session to which it belongs—and creates a new session table entry for it.
- **Uninterrupted Sessions**—If you reset the device or even change a component in the core section of a policy and SYN checking is enabled, all existing sessions or those sessions to which the policy change applies are disrupted and must be restarted. Disabling SYN checking avoids such disruptions to network traffic flows.



**NOTE:** A solution to this scenario is to install the device with SYN checking disabled initially. Then, after a few hours—when established sessions are running through the device—enable SYN checking. The core section in a policy contains the following main components: source and destination zones, source and destination addresses, one or more services, and an action.

However, the previous advantages exact the following security sacrifices:

- **Reconnaissance Holes**—When an initial TCP segment with a non-SYN flag—such as ACK, URG, RST, FIN—arrives at a closed port, many operating systems (Windows, for example) respond with a TCP segment that has the RST flag set. If the port is open, then the recipient does not generate any response.

By analyzing these responses or lack thereof, an intelligence gatherer can perform reconnaissance on the protected network and also on the Junos OS policy set. If a TCP segment is sent with a non-SYN flag set and the policy permits it through, the destination host receiving such a segment might drop it and respond with a TCP segment that has the RST flag set. Such a response informs the perpetrator of the presence of an active host at a specific address and that the targeted port number is closed. The intelligence gatherer also learns that the firewall policy permits access to that port number on that host.

By enabling SYN flag checking, Junos OS drops TCP segments without a SYN flag if they do not belong to an existing session. It does not return a TCP RST segment. Consequently, the scanner gets no replies regardless of the policy set or whether the port is open or closed on the targeted host.

- **Session Table Floods**—If SYN checking is disabled, an attacker can bypass the Junos OS SYN flood protection feature by flooding a protected network with a barrage of TCP segments that have non-SYN flags set. Although the targeted hosts drop the packets—and possibly send TCP RST segments in reply—such a flood can fill up the session table of the Juniper Networks device. When the session table is full, the device cannot process new sessions for legitimate traffic.

By enabling SYN checking and SYN flood protection, you can thwart this kind of attack. Checking that the SYN flag is set on the initial packet in a session forces all new sessions to begin with a TCP segment that has the SYN flag set. SYN flood protection then limits the number of TCP SYN segments per second so that the session table does not become overwhelmed.

If you do not need SYN checking disabled, Juniper Networks strongly recommends that it be enabled (its default state for an initial installation of Junos OS). You can enable it with the **set flow tcp-syn-check** command. With SYN checking enabled, the device rejects TCP segments with non-SYN flags set unless they belong to an established session.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Setting TCP SYN Checking (CLI Procedure)

With SYN checking enabled, the device rejects TCP segments with non-SYN flags set unless they belong to an established session. Enabling SYN checking can help prevent attacker reconnaissance and session table floods. TCP SYN checking is enabled by default.

To disable SYN checking:

```
user@host#set security flow tcp-session no-syn-check
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*



### Setting Strict SYN Checking (CLI Procedure)

With strict SYN checking enabled, the device enables the strict three-way handshake check for the TCP session. It enhances security by dropping data packets before the three-way handshake is done. TCP strict SYN checking is disabled by default.



**NOTE:** The `strict-syn-check` option cannot be enabled if `no-syn-check` or `no-syn-check-in-tunnel` is enabled.

To enable strict SYN checking:

```
user@host#set security flow tcp-session strict-syn-check
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## IP Spoofing

- Understanding IP Spoofing on page 729
- Example: Blocking IP Spoofing on page 729

### Understanding IP Spoofing

One method of attempting to gain access to a restricted area of the network is to insert a false source address in the packet header to make the packet appear to come from a trusted source. This technique is called IP spoofing. The mechanism to detect IP spoofing relies on route table entries. For example, if a packet with source IP address 10.1.1.6 arrives at ge-0/0/1, but Junos OS has a route to 10.1.1.0/24 through ge-0/0/0, a check for IP spoofing discovers that this address arrived at an invalid interface as defined in the route table. A valid packet from 10.1.1.6 can only arrive via ge-0/0/0, not ge-0/0/1. Therefore, Junos OS concludes that the packet has a spoofed source IP address and discards it.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Blocking IP Spoofing

This example shows how to configure a screen to block IP spoof attacks.

- Requirements on page 729
- Overview on page 729
- Configuration on page 730
- Verification on page 730

#### Requirements

Before you begin, understand how IP Spoofing works. See “Understanding IP Spoofing” on page 729.

#### Overview

One method of attempting to gain access to a restricted area of a network is to insert a bogus source address in the packet header to make the packet appear to come from a trusted source. This technique is called IP spoofing.

In this example, you configure a screen called screen-1 to block IP spoof attacks and enable the screen in the zone-1 security zone.

### Configuration

#### Step-by-Step Procedure

To block IP spoofing:

1. Configure the screen.

```
[edit]
user@host# set security screen ids-option screen-1 ip spoofing
```

2. Enable the screen in the security zone.

```
[edit]
user@host# set security zone security-zone zone-1 screen screen-1
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security screen** command.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

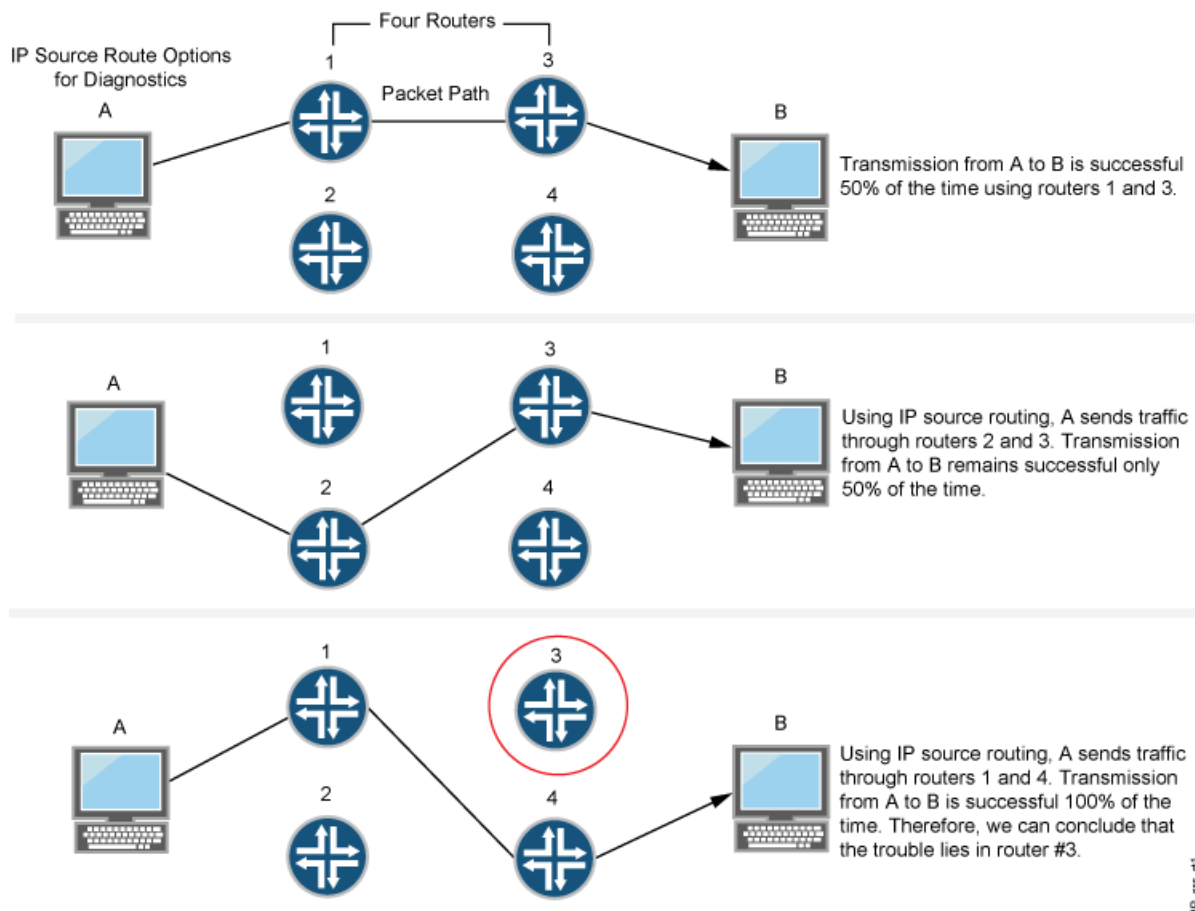
## IP Source Route Options

- Understanding IP Source Route Options on page 730
- Example: Blocking Packets with Either a Loose or a Strict Source Route Option Set on page 732
- Example: Detecting Packets with Either a Loose or a Strict Source Route Option Set on page 733

### Understanding IP Source Route Options

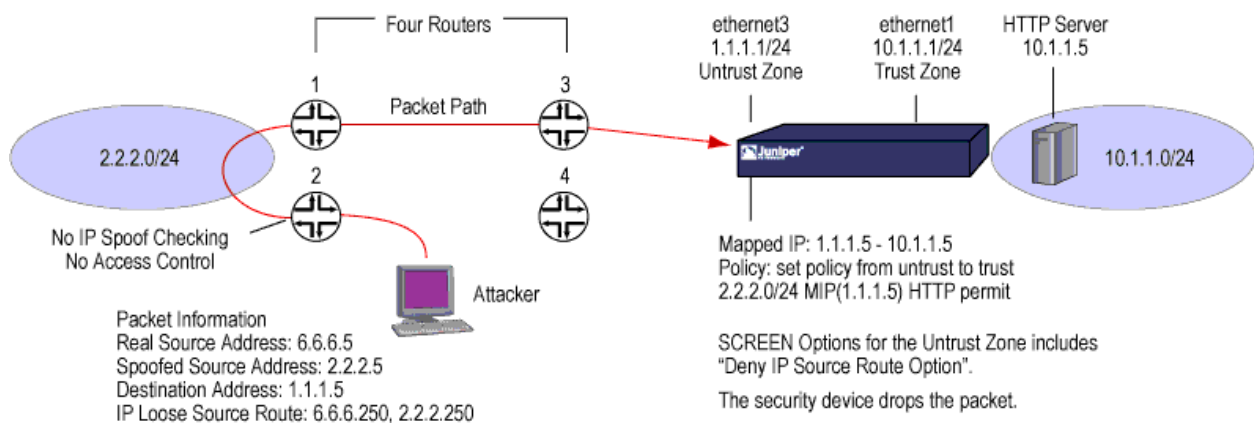
Source routing was designed to allow users at the source of an IP packet transmission to specify the IP addresses of the devices (also referred to as “hops” ) along the path that they want an IP packet to take on its way to its destination. The original intent of the IP source route options was to provide routing control tools to aid diagnostic analysis. If, for example, the transmission of a packet to a particular destination meets with irregular success, you might first use either the record route or the timestamp IP option to discover the addresses of devices along the path or paths that the packet takes. You can then use either the loose or the strict source route option to direct traffic along a specific path, using the addresses you learned from the results that the record route or timestamp options produced. By changing device addresses to alter the path and sending several packets along different paths, you can note changes that either improve or lessen the success rate. Through analysis and the process of elimination, you might be able to deduce where the trouble lies. See Figure 64 on page 731.

Figure 64: IP Source Routing



Although the uses of IP source route options were originally benign, attackers have learned to put them to more devious uses. They can use IP source route options to hide their true address and access restricted areas of a network by specifying a different path. For an example showing how an attacker can put both deceptions to use, consider the following scenario as illustrated in Figure 65 on page 731.

Figure 65: Loose IP Source Route Option for Deception



Junos OS only allows traffic 2.2.2.0/24 if it comes through ethernet1, an interface bound to zone\_external. Devices 3 and 4 enforce access controls but devices 1 and 2 do not. Furthermore, device 2 does not check for IP spoofing. The attacker spoofs the source address and, by using the loose source route option, directs the packet through device 2 to the 2.2.2.0/24 network and from there out device 1. Device 1 forwards it to device 3, which forwards it to the Juniper Networks device. Because the packet came from the 2.2.2.0/24 subnet and has a source address from that subnet, it seems to be valid. However, one remnant of the earlier chicanery remains: the loose source route option. In this example, you have enabled the deny IP source route screen option for zone\_external. When the packet arrives at ethernet3, the device rejects it.

You can enable the device to either block any packets with loose or strict source route options set or detect such packets and then record the event in the counters list for the ingress interface. The screen options are as follows:

- **Deny IP Source Route Option**—Enable this option to block all IP traffic that employs the loose or strict source route option. Source route options can allow an attacker to enter a network with a false IP address.
- **Detect IP Loose Source Route Option**—The device detects packets where the IP option is 3 (Loose Source Routing) and records the event in the screen counters list for the ingress interface. This option specifies a partial route list for a packet to take on its journey from source to destination. The packet must proceed in the order of addresses specified, but it is allowed to pass through other devices in between those specified.
- **Detect IP Strict Source Route Option**—The device detects packets where the IP option is 9 (Strict Source Routing) and records the event in the screen counters list for the ingress interface. This option specifies the complete route list for a packet to take on its journey from source to destination. The last address in the list replaces the address in the destination field. Currently, this screen option is applicable to IPv4 only.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Blocking Packets with Either a Loose or a Strict Source Route Option Set

This example shows how to block packets with either a loose or a strict source route option set.

- Requirements on page 732
- Overview on page 732
- Configuration on page 733
- Verification on page 733

#### Requirements

Before you begin, understand how IP source route options work. See “Understanding IP Source Route Options” on page 730.

#### Overview

Source routing allows users at the source of an IP packet transmission to specify the IP addresses of the devices (also referred to as “hops” ) along the path that they want an

IP packet to take on its way to its destination. The original intent of the IP source route options was to provide routing control tools to aid diagnostic analysis.

You can enable the device to either block any packets with loose or strict source route options set or detect such packets and then record the event in the counters list for the ingress interface.

In this example you create the screen called screen-1 to block packets with either a loose or a strict source route option set and enable the screen in the zone-1 security zone.

### Configuration

#### Step-by-Step Procedure

To block packets with either the loose or the strict source route option set:

1. Configure the screen.  

```
[edit]
user@host# set security screen ids-option screen-1 ip source-route-option.
```
2. Enable the screen in the security zone.  

```
[edit]
user@host# set security zones security-zone zone-1 screen ip-filter-src
```
3. If you are done configuring the device, commit the configuration.  

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security screen** command.

#### Example: Detecting Packets with Either a Loose or a Strict Source Route Option Set

This example shows how to detect packets with either a loose or a strict source route option set.

- Requirements on page 733
- Overview on page 733
- Configuration on page 734
- Verification on page 734

### Requirements

Before you begin, understand how IP source route options work. See “Understanding IP Source Route Options” on page 730.

### Overview

Source routing allows users at the source of an IP packet transmission to specify the IP addresses of the devices (also referred to as “hops” ) along the path that they want an IP packet to take on its way to its destination. The original intent of the IP source route options was to provide routing control tools to aid diagnostic analysis.

You can enable the device to either block any packets with loose or strict source route options set or detect such packets and then record the event in the counters list for the ingress interface.

In this example, you create two screens called screen-1 and screen-2 to detect and record, but not block, packets with a loose or strict source route option set and enable the screens in the zone-1 security screen.

### Configuration

#### Step-by-Step Procedure

To detect and record, but not block, packets with a loose or strict source route option set:

1. Configure the loose source screen.

```
[edit]
user@host# set security screen ids-option screen-1 ip loose-source-route-option
```

2. Configure the strict source route screen.

```
[edit]
user@host# set security screen ids-option screen-2 ip strict-source-route-option
```



---

**NOTE:** Currently, this screen option supports IPv4 only.

---

3. Enable the screens in the zone-1 security zone.

```
[edit]
user@host# set security zones security-zone zone-1 screen screen-2
```

4. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

### Verification

To verify the configuration is working properly, enter the **show security screen** command.

## CHAPTER 37

# Suspicious Packet Attributes

- Suspicious Packet Attributes Overview on page 735
- ICMP Fragment Protection on page 735
- Large ICMP Packet Protection on page 736
- Bad IP Option Protection on page 738
- Unknown Protocol Protection on page 739
- IP Packet Fragment Protection on page 740
- SYN Fragment Protection on page 741

## Suspicious Packet Attributes Overview

---

Attackers can craft packets to perform reconnaissance or launch denial-of-service (DoS) attacks. Sometimes it is unclear what the intent of a crafted packet is, but the very fact that it is crafted suggests that it is being put to some kind of insidious use.

The following topics describe screen options that block suspicious packets that might contain hidden threats:

- Understanding ICMP Fragment Protection on page 736
- Understanding Large ICMP Packet Protection on page 737
- Understanding Bad IP Option Protection on page 738
- Understanding Unknown Protocol Protection on page 739
- Understanding IP Packet Fragment Protection on page 740
- Understanding SYN Fragment Protection on page 741

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## ICMP Fragment Protection

---

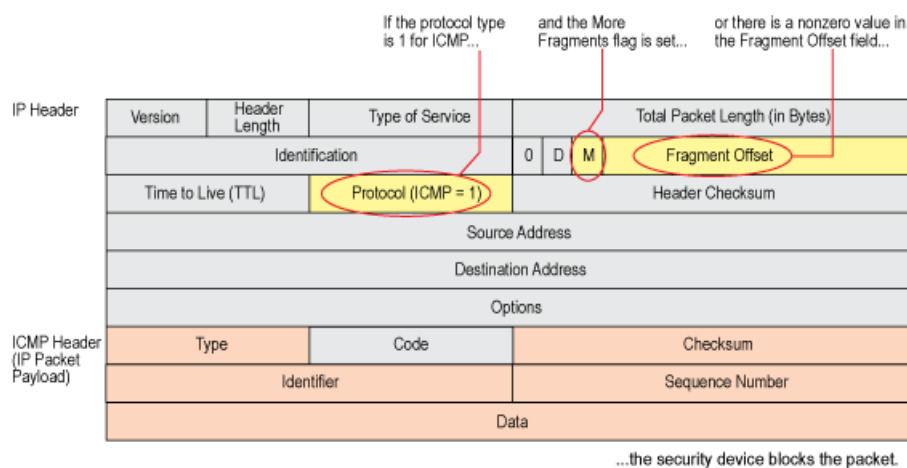
- Understanding ICMP Fragment Protection on page 736
- Example: Blocking Fragmented ICMP Packets (CLI) on page 736

## Understanding ICMP Fragment Protection

Internet Control Message Protocol (ICMP) provides error reporting and network probe capabilities. Because ICMP packets contain very short messages, there is no legitimate reason for ICMP packets to be fragmented. If an ICMP packet is so large that it must be fragmented, something is amiss.

When you enable the ICMP fragment protection screen option, Junos OS blocks any ICMP packet that has the More Fragments flag set or that has an offset value indicated in the offset field. See Figure 66 on page 736.

Figure 66: Blocking ICMP Fragments



**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Example: Blocking Fragmented ICMP Packets (CLI)

The following example shows how to configure the **icmp-fragment** screen to block fragmented ICMP packets originating from the **zone** security zone.

To block fragmented ICMP packets:

1. Configure the **icmp-fragment** screen:
 

```
user@host# set security screen ids-option icmp-fragment icmp fragment
```
2. Configure the **zone** security zone:
 

```
user@host# set security zones security-zone zone screen icmp-fragment
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Large ICMP Packet Protection

- Understanding Large ICMP Packet Protection on page 737
- Example: Blocking Large ICMP Packets (CLI) on page 737

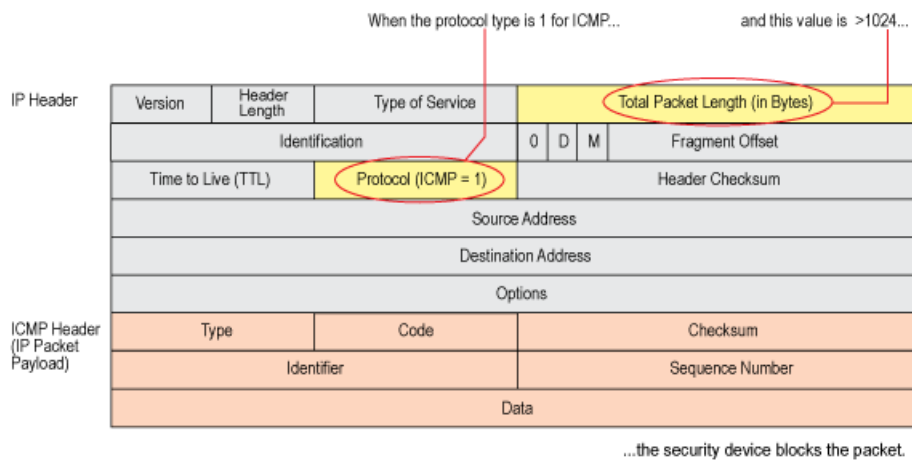


## Understanding Large ICMP Packet Protection

Internet Control Message Protocol (ICMP) provides error reporting and network probe capabilities. Because ICMP packets contain very short messages, there is no legitimate reason for large ICMP packets. If an ICMP packet is unusually large, something is amiss.

For example, the SRX 210 uses ICMP as a channel for transmitting covert messages. The presence of large ICMP packets might expose a compromised machine acting as a SRX 210 agent. It also might indicate some other kind of questionable activity. See Figure 67 on page 737.

**Figure 67: Blocking Large ICMP Packets**



When you enable the large size ICMP packet protection screen option, Junos OS drops ICMP packets with a length greater than 1024 bytes.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Example: Blocking Large ICMP Packets (CLI)

The following example shows how to configure the **icmp-large** screen to block large ICMP packets originating from the **zone** security zone.

To block large ICMP packets:

1. Configure the **icmp-large** screen:
 

```
user@host# set security screen ids-option icmp-large icmp large
```
2. Configure the **zone** security zone:
 

```
user@host# set security zones security-zone zone screen icmp-large
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Bad IP Option Protection

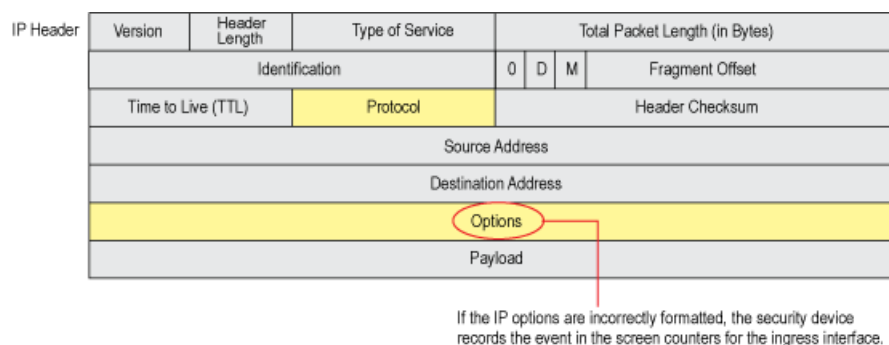
- Understanding Bad IP Option Protection on page 738
- Example: Blocking IP Packets with Incorrectly Formatted Options (CLI) on page 738

### Understanding Bad IP Option Protection

The IP standard RFC 791, *Internet Protocol*, specifies a set of eight options that provide special routing controls, diagnostic tools, and security. Although the original, intended uses for these options served worthy ends, people have figured out ways to twist these options to accomplish less commendable objectives.

Either intentionally or accidentally, attackers sometimes configure IP options incorrectly, producing either incomplete or malformed fields. Regardless of the intentions of the person who crafted the packet, the incorrect formatting is anomalous and potentially harmful to the intended recipient. See Figure 68 on page 738.

**Figure 68: Incorrectly Formatted IP Options**



When you enable the bad IP option protection screen option, Junos OS blocks packets when any IP option in the IP packet header is incorrectly formatted. Additionally, Junos OS records the event in the event log.

### Example: Blocking IP Packets with Incorrectly Formatted Options (CLI)

The following example shows how to configure the **ip-bad-option** screen to block large ICMP packets originating from the **zone** security zone.

To detect and block IP packets with incorrectly formatted IP options:

1. Configure the **ip-bad-option** screen:

```
user@host# set security screen ids-option ip-bad-option ip bad-option
```



**NOTE:** Currently this screen option is applicable only to IPv4.

2. Configure the **zone** security zone:

```
user@host# set security zones security-zone zone screen ip-bad-option
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Unknown Protocol Protection

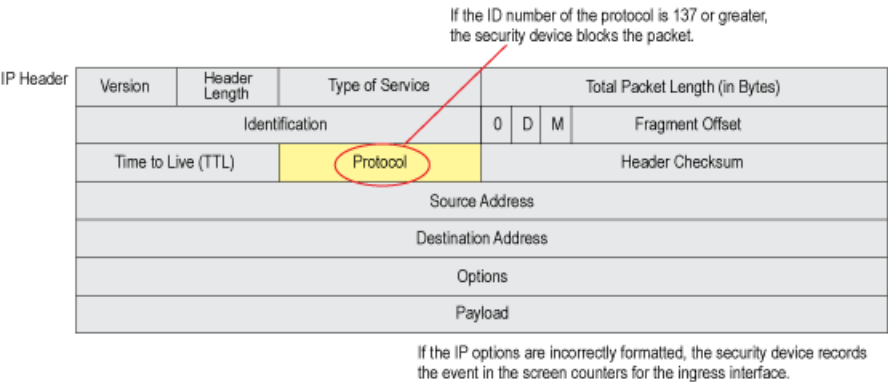
- Understanding Unknown Protocol Protection on page 739
- Example: Dropping Packets Using an Unknown Protocol (CLI) on page 739

Understanding Unknown Protocol Protection

Based on RFC 1700, the protocol types with ID numbers of 137 or greater are reserved and undefined at this time. Precisely because these protocols are undefined, there is no way to know in advance if a particular unknown protocol is benign or malicious.

Unless your network makes use of a nonstandard protocol with an ID number of 137 or greater, a cautious stance is to block such unknown elements from entering your protected network. See Figure 69 on page 739.

Figure 69: Unknown Protocols



When you enable the unknown protocol protection screen option, Junos OS drops packets when the protocol field contains a protocol ID number of 137 or greater by default.



**NOTE:** When you enable the unknown protocol protection screen option for IPv6 protocol, Junos OS drops packets when the protocol field contains a protocol ID number of 139 or greater by default.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Dropping Packets Using an Unknown Protocol (CLI)

The following example shows how to configure the **unknown-protocol** screen to block packets with an unknown protocol originating from the **zone** security zone.

To drop packets that use an unknown protocol:

1. Configure the **unknown-protocol** screen:

```
user@host# set security screen ids-option unknown-protocol ip unknown-protocol
```

2. Configure the **zone** security zone:

```
user@host# set security zones security-zone zone screen unknown-protocol
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

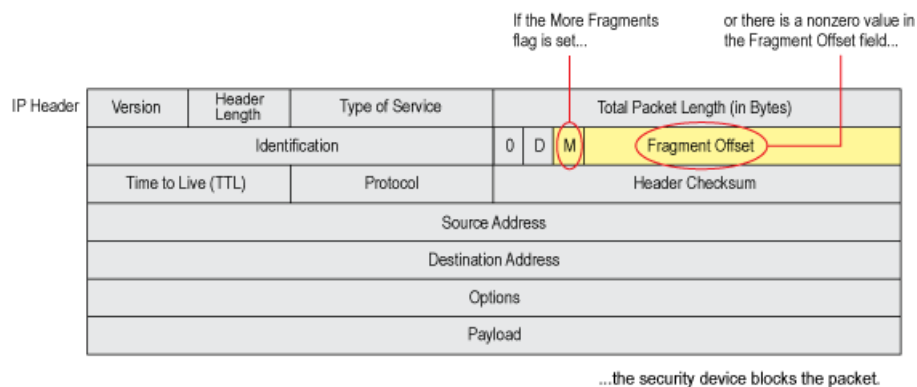
## IP Packet Fragment Protection

- Understanding IP Packet Fragment Protection on page 740
- Example: Dropping Fragmented IP Packets (CLI) on page 740

### Understanding IP Packet Fragment Protection

As packets traverse different networks, it is sometimes necessary to break a packet into smaller pieces (fragments) based upon the maximum transmission unit (MTU) of each network. IP fragments might contain an attacker's attempt to exploit the vulnerabilities in the packet reassembly code of specific IP stack implementations. When the victim receives these packets, the results can range from processing the packets incorrectly to crashing the entire system. See Figure 70 on page 740.

**Figure 70: IP Packet Fragments**



When you enable Junos OS to deny IP fragments on a security zone, it blocks all IP packet fragments that it receives at interfaces bound to that zone.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Dropping Fragmented IP Packets (CLI)

The following example shows how to configure the **block-frag** screen to drop fragmented IP packets originating from the **zone** security zone.

To drop fragmented IP packets:

1. Configure the **block-frag** screen:

```
user@host# set security screen ids-option block-frag ip block-frag
```

2. Configure the **zone** security zone:

```
user@host# set security zones security-zone zone screen block-frag
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## SYN Fragment Protection

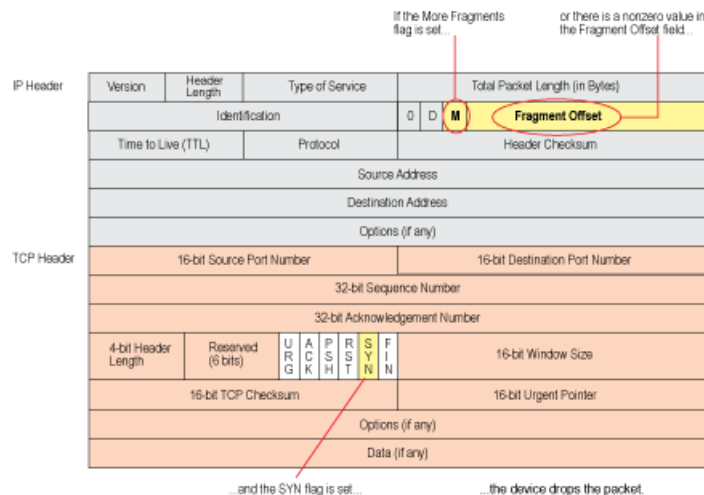
- Understanding SYN Fragment Protection on page 741
- Example: Dropping IP Packets Containing SYN Fragments (CLI) on page 742

### Understanding SYN Fragment Protection

The IP encapsulates a TCP SYN segment in the IP packet that initiates a TCP connection. Because the purpose of this packet is to initiate a connection and invoke a SYN/ACK segment in response, the SYN segment typically does not contain any data. Because the IP packet is small, there is no legitimate reason for it to be fragmented.

A fragmented SYN packet is anomalous, and, as such, it is suspect. To be cautious, block such unknown elements from entering your protected network. See Figure 71 on page 741.

**Figure 71: SYN Fragments**



When you enable the SYN fragment detection screen option, Junos OS detects packets when the IP header indicates that the packet has been fragmented and the SYN flag is set in the TCP header. Junos OS records the event in the screen counters list for the ingress interface.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Dropping IP Packets Containing SYN Fragments (CLI)

The following example shows how to configure the **syn-frag** screen to drop fragmented SYN packets originating from the **zone** security zone.

To drop IP packets containing SYN fragments:

1. Configure the **syn-frag** screen:

```
user@host# set security screen ids-option syn-frag tcp syn-frag
```

2. Configure the **zone** security zone:

```
user@host# set security zones security-zone zone screen syn-frag
```

**Related Topics**   • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## CHAPTER 38

# Denial-of-Service Attacks

- DoS Attack Overview on page 743
- Firewall DoS Attacks on page 743
- Network DoS Attacks on page 748
- OS-Specific DoS Attacks on page 761

### DoS Attack Overview

---

The intent of a denial-of-service (DoS) attack is to overwhelm the targeted victim with a tremendous amount of bogus traffic so that the victim becomes so preoccupied processing the bogus traffic that legitimate traffic cannot be processed. The target can be the firewall, the network resources to which the firewall controls access, or the specific hardware platform or operating system of an individual host.

If a DoS attack originates from multiple source addresses, it is known as a distributed denial-of-service (DDoS) attack. Typically, the source address of a DoS attack is spoofed. The source addresses in a DDoS attack might be spoofed, or the actual addresses of compromised hosts might be used as “zombie agents” to launch the attack.

The device can defend itself and the resources it protects from DoS and DDoS attacks.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Firewall DoS Attacks

---

- Firewall DoS Attacks Overview on page 743
- Session Table Flood Attacks on page 744
- SYN-ACK-ACK Proxy Flood Attacks on page 747

### Firewall DoS Attacks Overview

The intent of a denial-of-service (DoS) attack is to overwhelm the targeted victim with a tremendous amount of bogus traffic so that the victim becomes so preoccupied processing the bogus traffic that legitimate traffic cannot be processed.

If attackers discover the presence of the Juniper Networks firewall, they might launch a DoS attack against it instead of the network behind it. A successful DoS attack against

a firewall amounts to a successful DoS attack against the protected network in that it thwarts attempts of legitimate traffic to traverse the firewall.

An attacker might use session table floods and SYN-ACK-ACK proxy floods to fill up the session table of Junos OS and thereby produce a DoS.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Session Table Flood Attacks

- Understanding Session Table Flood Attacks on page 744
- Understanding Source-Based Session Limits on page 744
- Example: Setting Source-Based Session Limits (CLI) on page 745
- Understanding Destination-Based Session Limits on page 746
- Example: Setting Destination-Based Session Limits (CLI) on page 746

### Understanding Session Table Flood Attacks

A successful DoS attack overwhelms its victim with such a massive barrage of false simulated traffic that it becomes unable to process legitimate connection requests. DoS attacks can take many forms—SYN flood, SYN-ACK-ACK flood, UDP flood, ICMP flood, and so on—but they all seek the same objective, which is to fill up their victim's session table.

When the session table is full, that host cannot create any new sessions and begins rejecting new connection requests. The source-based session limits screen option and the destination-based session limit screen option help mitigate such attacks.

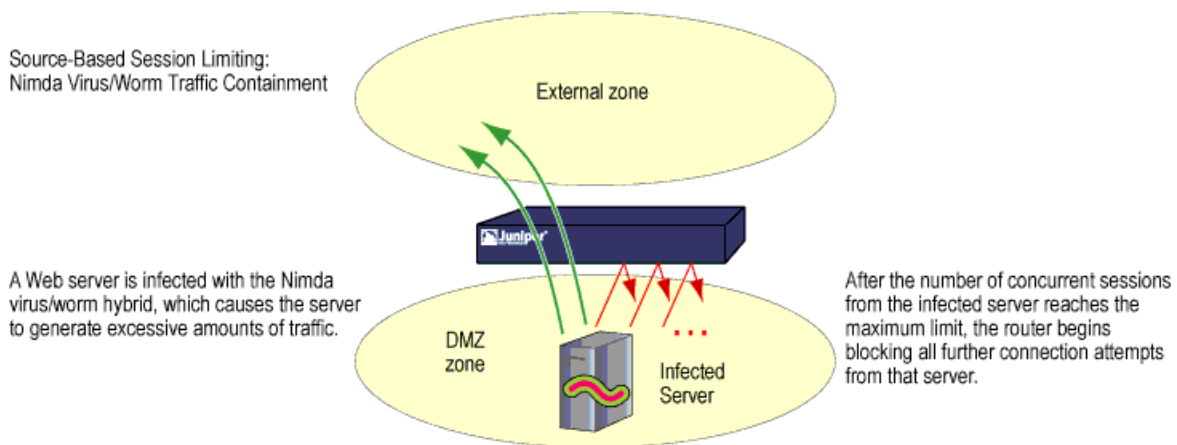
**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Understanding Source-Based Session Limits

In addition to limiting the number of concurrent sessions from the same source IP address, you can also limit the number of concurrent sessions to the same destination IP address. One benefit of setting a source-based session limit is that it can stem an attack such as the Nimda virus (which is actually both a virus and a worm) that infects a server and then begins generating massive amounts of traffic from that server. Because all the virus-generated traffic originates from the same IP address, a source-based session limit ensures that the firewall can curb such excessive amounts of traffic. See Figure 72 on page 745.



Figure 72: Limiting Sessions Based on Source IP Address



Another benefit of source-based session limiting is that it can mitigate attempts to fill up the firewall's session table if all the connection attempts originate from the same source IP address.

Determining what constitutes an acceptable number of connection requests requires a period of observation and analysis to establish a baseline for typical traffic flows. You also need to consider the maximum number of concurrent sessions required to fill up the session table of the particular Juniper Networks platform you are using. To see the maximum number of sessions that your session table supports, use the CLI command **get session**, and then look at the first line in the output, which lists the number of current (allocated) sessions, the maximum number of sessions, and the number of failed session allocations:

```
alloc 420/max 128000, alloc failed 0
```

The default maximum for source-based session limits is 128 concurrent sessions, a value that might need adjustment to suit the needs of your network environment and the platform.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Setting Source-Based Session Limits (CLI)

The following example shows you how to limit the amount of sessions that any one server in the DMZ and zone\_a zones can initiate. Because the DMZ zone only contains web servers, none of which should initiate traffic, you set the source-session limit at the lowest possible value: 1 session. On the other hand, the zone\_a zone contains personal computers, servers, printers, and so on, many of which do initiate traffic. For the zone\_a zone, you set the source-session limit maximum to 80 concurrent sessions.

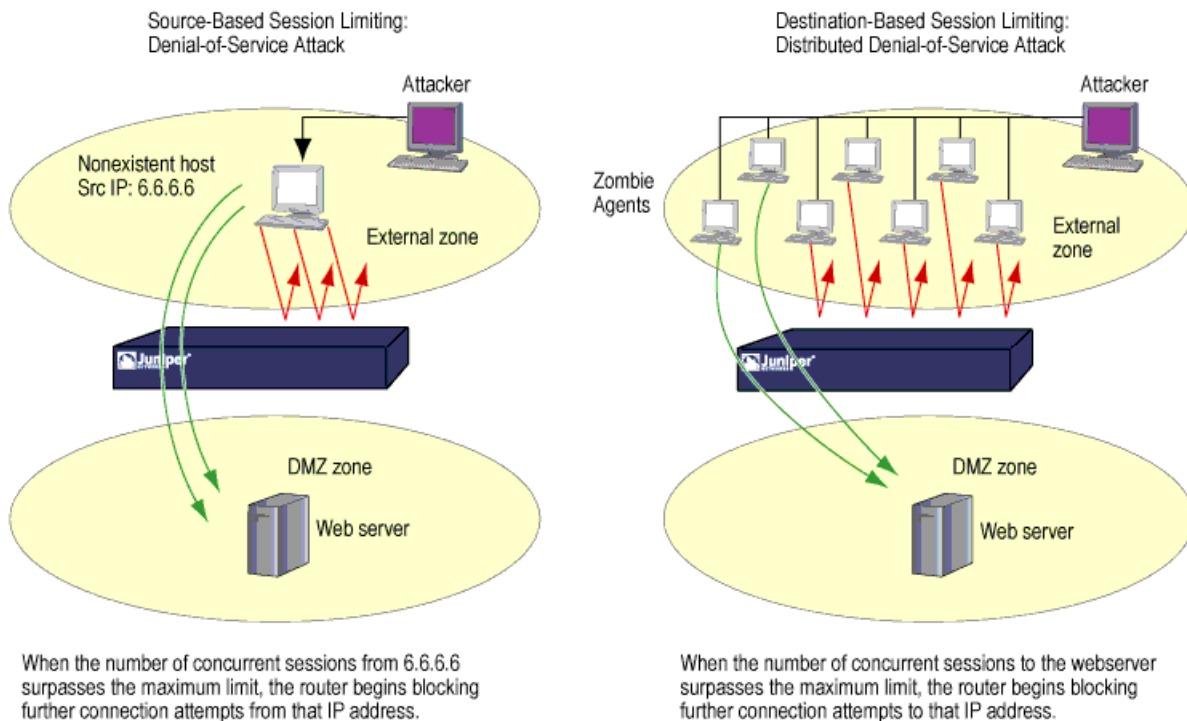
```
user@host# set security screen ids-option 1-limit-session limit-session source-ip-based 1
user@host# set security screen ids-option 100-limit-session limit-session source-ip-based 100
user@host# set security screen ids-option 80-limit-session limit-session source-ip-based 80
user@host# set security zones security-zone dmz screen 100-limit-session
user@host# set security zones security-zone zone_a screen 100-limit-session
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Understanding Destination-Based Session Limits

In addition to limiting the number of concurrent sessions from the same source IP address, you can also limit the number of concurrent sessions to the same destination IP address. A wily attacker can launch a distributed denial-of-service (DDoS) attack. In a DDoS attack, the malicious traffic can come from hundreds of hosts, known as “zombie agents,” that are surreptitiously under the control of an attacker. In addition to the SYN, UDP, and ICMP flood detection and prevention screen options, setting a destination-based session limit can ensure that Junos OS allows only an acceptable number of concurrent connection requests—no matter what the source—to reach any one host. See Figure 73 on page 746.

**Figure 73: Distributed DOS Attack**



The default maximum for destination-based session limits is 128 concurrent sessions, a value that might need adjustment to suit the needs of your network environment and the platform.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Setting Destination-Based Session Limits (CLI)

The following example shows you how to limit the amount of traffic to a webserver at 1.2.2.5. The server is in the DMZ zone. The example assumes that after observing the traffic flow from the external zone to this server for a month, you have determined that the average number of concurrent sessions it receives is 2000.

The example shows how to set the new session limit at 4000 concurrent sessions. Although traffic spikes might sometimes exceed that limit, the example assumes that you are opting for firewall security over occasional server inaccessibility.

```
user@host# set security screen ids-option 4000-limit-session limit-session
destination-ip-based 4000
user@host# set security screen ids-option 100-limit-session limit-session
destination-ip-based 100
user@host# set security zones security-zone external_zone screen 100-limit-session
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## SYN-ACK-ACK Proxy Flood Attacks

- Understanding SYN-ACK-ACK Proxy Flood Attacks on page 747
- Example: Protecting Against a SYN-ACK-ACK Proxy Flood Attack (CLI) on page 747

### Understanding SYN-ACK-ACK Proxy Flood Attacks

When an authentication user initiates a Telnet or an FTP connection, the user sends a SYN segment to the Telnet or FTP server. Junos OS intercepts the SYN segment, creates an entry in its session table, and proxies a SYN-ACK segment to the user. The user then replies with an ACK segment. At this point, the initial three-way handshake is complete. Junos OS sends a login prompt to the user. If the user, with malicious intent, does not log in but instead continues initiating SYN-ACK-ACK sessions, the firewall session table can fill up to the point where the device begins rejecting legitimate connection requests.

To prevent such an attack, you can enable the SYN-ACK-ACK proxy protection screen option. After the number of connections from the same IP address reaches the SYN-ACK-ACK proxy threshold, Junos OS rejects further connection requests from that IP address. By default, the threshold is 512 connections from any single IP address. You can change this threshold (to any number between 1 and 250,000) to better suit the requirements of your network environment.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Protecting Against a SYN-ACK-ACK Proxy Flood Attack (CLI)

The following example shows you how to enable protection against a SYN-ACK-ACK proxy flood. (The value unit is connections per source address. The default value is 512 connections from any single address.) In the example, the specified zone is where the attack originated.

```
user@host# set security screen ids-option 1000-syn-ack-ack-proxy tcp syn-ack-ack-proxy
threshold 1000
user@host# set security zones security-zone zone screen 1000-syn-ack-ack-proxy
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Network DoS Attacks

---

- Network DoS Attacks Overview on page 748
- SYN Flood Attacks on page 748
- SYN Cookie Protection on page 756
- ICMP Flood Protection on page 758
- UDP Flood Attacks on page 759
- Land Attacks on page 760

### Network DoS Attacks Overview

A denial-of-service (DoS) attack directed against one or more network resources floods the target with an overwhelming number of SYN, ICMP, or UDP packets or with an overwhelming number of SYN fragments.

Depending on the attackers' purpose and the extent and success of previous intelligence gathering efforts, the attackers might single out a specific host, such as a device or server or they might aim at random hosts across the targeted network. Either approach has the potential of upsetting service to a single host or to the entire network, depending on how critical the role of the victim is to the rest of the network.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### SYN Flood Attacks

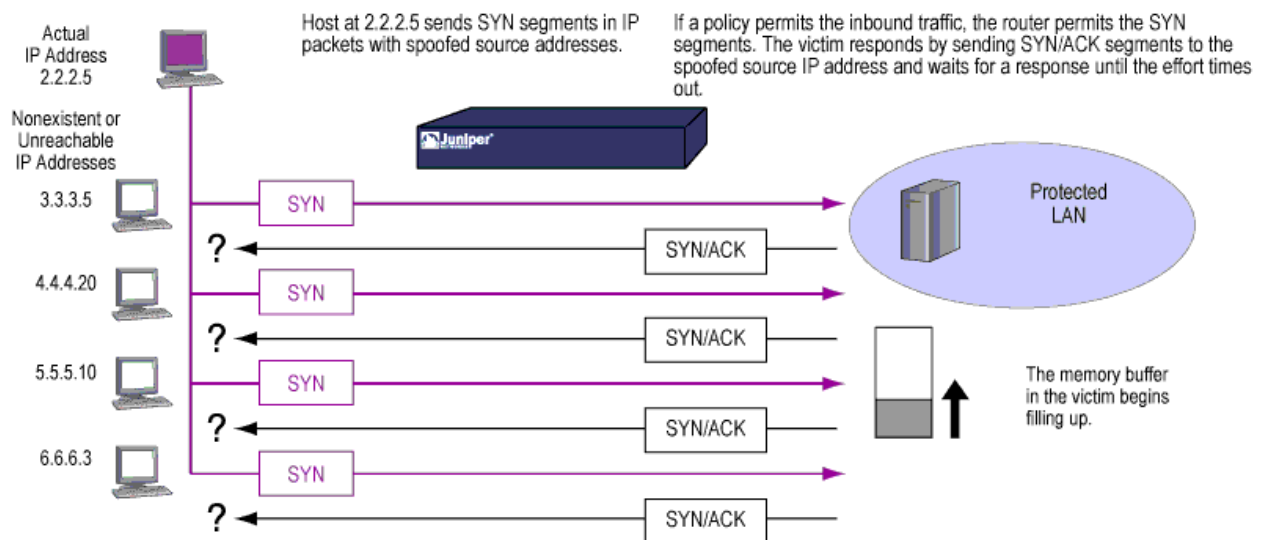
- Understanding SYN Flood Attacks on page 748
- Example: Enabling SYN Flood Protection (CLI) on page 753
- Configuring SYN Flood Protection Options (CLI Procedure) on page 753
- Example: Enabling SYN Flood Protection for Webservers in the DMZ (CLI) on page 753

#### Understanding SYN Flood Attacks

A SYN flood occurs when a host becomes so overwhelmed by SYN segments initiating incomplete connection requests that it can no longer process legitimate connection requests.

Two hosts establish a TCP connection with a triple exchange of packets known as a *three-way handshake*: A sends a SYN segment to B; B responds with a SYN/ACK segment; and A responds with an ACK segment. A SYN flood attack inundates a site with SYN segments containing forged (spoofed) IP source addresses with nonexistent or unreachable addresses. B responds with SYN/ACK segments to these addresses and then waits for responding ACK segments. Because the SYN/ACK segments are sent to nonexistent or unreachable IP addresses, they never elicit responses and eventually time out. See Figure 74 on page 749.

Figure 74: SYN Flood Attack



By flooding a host with incomplete TCP connections, the attacker eventually fills the memory buffer of the victim. Once this buffer is full, the host can no longer process new TCP connection requests. The flood might even damage the victim's operating system. Either way, the attack disables the victim and its normal operations.

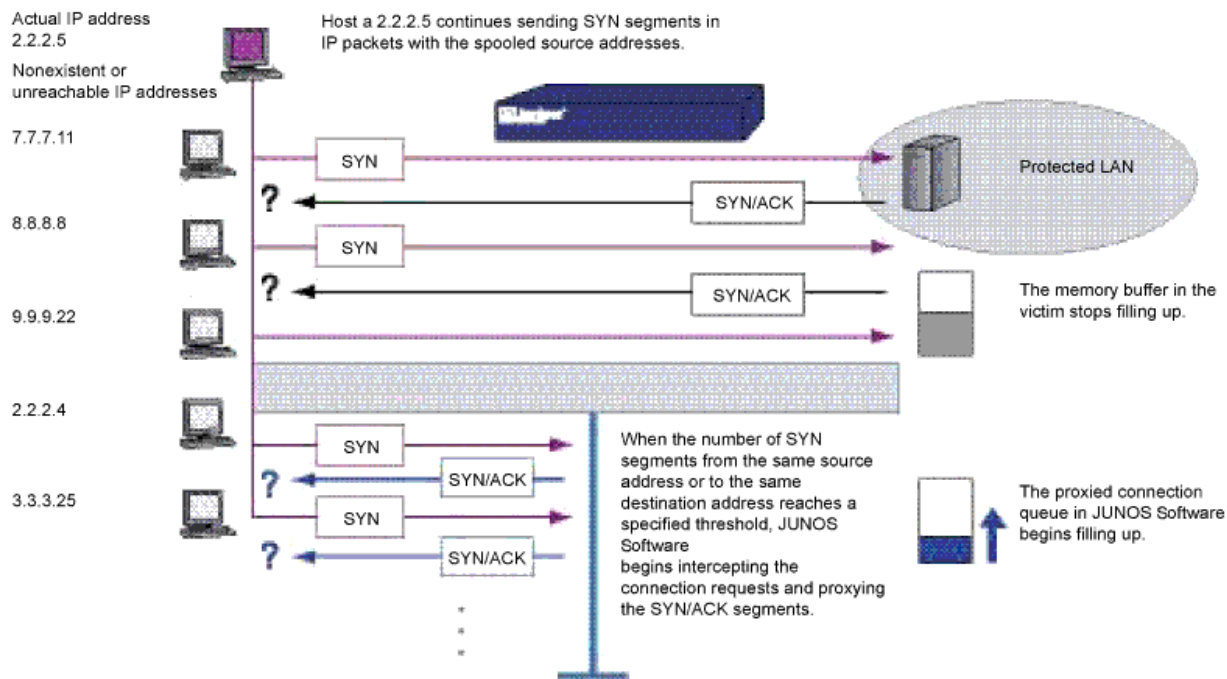
This topic includes the following sections:

- SYN Flood Protection on page 749
- SYN Flood Options on page 751

## SYN Flood Protection

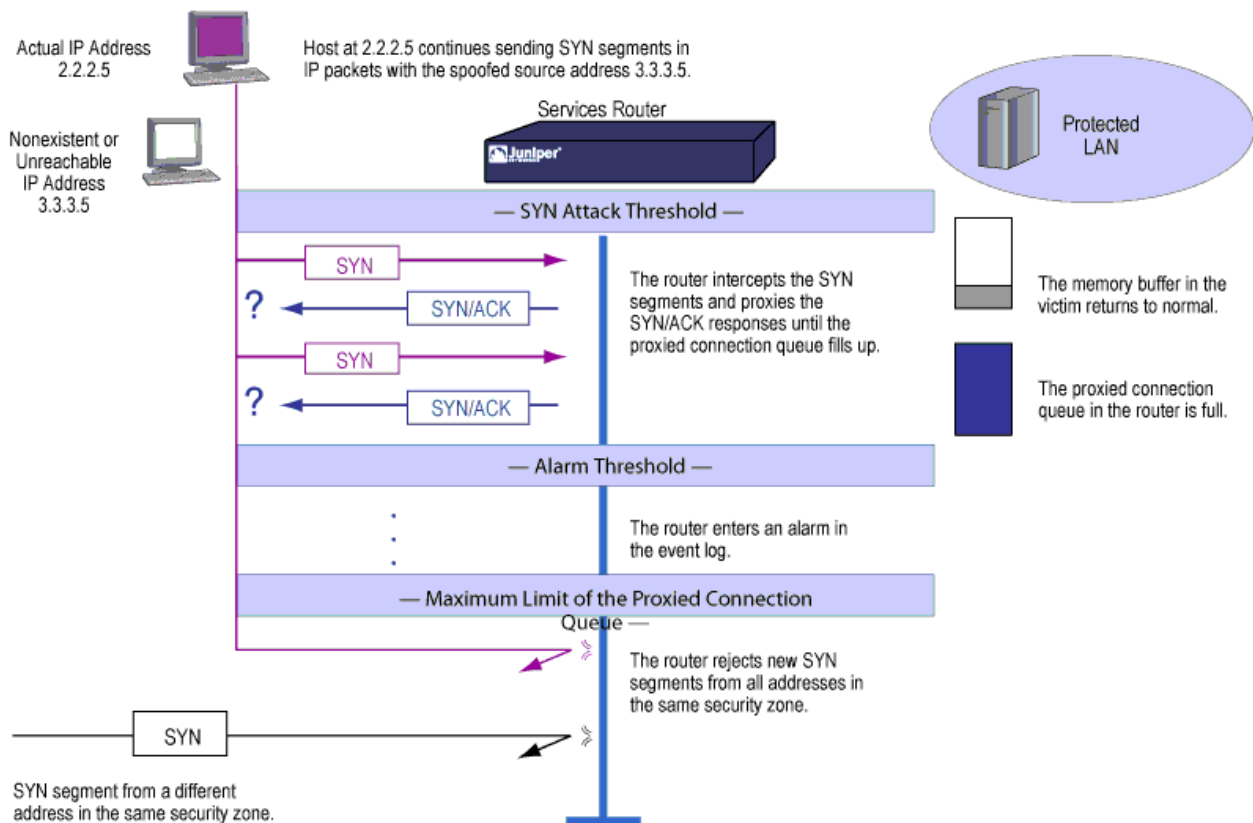
Junos OS can impose a limit on the number of SYN segments permitted to pass through the firewall per second. You can base the attack threshold on the destination address and port, the destination address only, or the source address only. When the number of SYN segments per second exceeds one of these thresholds, Junos OS starts proxying incoming SYN segments, replying with SYN/ACK segments and storing the incomplete connection requests in a connection queue. The incomplete connection requests remain in the queue until the connection is completed or the request times out. In Figure 75 on page 750, the SYN attack threshold has passed, and Junos OS has started proxying SYN segments.

Figure 75: Proxying SYN Segments



In Figure 76 on page 751, the proxied connection queue has completely filled up, and Junos OS is rejecting new incoming SYN segments. This action shields hosts on the protected network from the bombardment of incomplete three-way handshakes.

Figure 76: Rejecting New SYN Segments



The device starts receiving new SYN packets when the proxy queue drops below the maximum limit.



**NOTE:** The procedure of proxying incomplete SYN connections above a set threshold pertains only to traffic permitted by existing policies. Any traffic for which a policy does not exist is automatically dropped.

## SYN Flood Options

You can set the following parameters for proxying uncompleted TCP connection requests:

- **Attack Threshold**—This option allows you to set the number of SYN segments (that is, TCP segments with the SYN flag set) to the same destination address and port number per second required to activate the SYN proxying mechanism. Although you can set the threshold to any number, you need to know the normal traffic patterns at your site to set an appropriate threshold for it. For example, if it is an e-business site that normally gets 20,000 SYN segments per second, you might want to set the threshold to 30,000 per second. If a smaller site normally gets 20 SYN segments per second, you might consider setting the threshold to 40.
- **Alarm Threshold**—This option allows you to set the number of proxied, half-complete TCP connection requests per second after which Junos OS enters an alarm in the event

log. The value you set for an alarm threshold triggers an alarm when the number of proxied, half-completed connection requests to the same destination address and port number per second exceeds that value. For example, if you set the SYN attack threshold at 2000 SYN segments per second and the alarm at 1000, then a total of 3001 SYN segments to the same destination address and port number per second is required to trigger an alarm entry in the log. More precisely:

1. The firewall passes the first 2000 SYN segments per second that meet policy requirements.
2. The firewall proxies the next 1000 SYN segments in the same second.
3. The 1001st proxied connection request (or 3001st connection request in that second) triggers the alarm.

For each SYN segment to the same destination address and port number in excess of the alarm threshold, the attack detection module generates a message. At the end of the second, the logging module compresses all similar messages into a single log entry that indicates how many SYN segments to the same destination address and port number arrived after exceeding the alarm threshold. If the attack persists beyond the first second, the event log enters an alarm every second until the attack stops.

- **Source Threshold**—This option allows you to specify the number of SYN segments received per second from a single source IP address—regardless of the destination IP address and port number—before Junos OS begins dropping connection requests from that source.

Tracking a SYN flood by source address uses different detection parameters from tracking a SYN flood by destination address and destination port number. When you set a SYN attack threshold and a source threshold, you put both the basic SYN flood protection mechanism and the source-based SYN flood tracking mechanism in effect.

- **Destination Threshold**—This option allows you to specify the number of SYN segments received per second for a single destination IP address before Junos OS begins dropping connection requests to that destination. If a protected host runs multiple services, you might want to set a threshold based on destination IP address only—regardless of the destination port number.

When you set a SYN attack threshold and a destination threshold, you put both the basic SYN flood protection mechanism and the destination-based SYN flood tracking mechanism in effect.

Tracking a SYN flood by destination address uses different detection parameters from tracking a SYN flood by destination address and destination port number. Consider the following case where Junos OS has policies permitting FTP requests (port 21) and HTTP requests (port 80) to the same server. If the SYN flood attack threshold is 1000 packets per second (pps) and an attacker sends 999 FTP packets and 999 HTTP pps, neither set of packets (where a set is defined as having the same destination address and port number) activates the SYN proxying mechanism. The basic SYN flood attack mechanism tracks destination address and port number, and neither set exceeds the attack threshold of 1000 pps. However, if the destination threshold is 1000 pps, Junos



OS treats both FTP and HTTP packets with the same destination address as members of a single set and rejects the 1001st packet—FTP or HTTP—to that destination.

- **Timeout**—This option allows you to set the maximum length of time before a half-completed connection is dropped from the queue. The default is 20 seconds, and you can set the timeout from 0–50 seconds. You might try decreasing the timeout value to a shorter length until you begin to see any dropped connections during normal traffic conditions. Twenty seconds is a very conservative timeout for a three-way handshake ACK response.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Enabling SYN Flood Protection (CLI)

The following example shows you how to enable the SYN flood protection screen option and define its parameters. In the example, the specified zone is where a flood might originate.

```
user@host# set security screen zone-syn-flood tcp syn-flood timeout 20
user@host# set security zones security-zone zone screen zone-syn-flood
user@host# set zone zone screen syn-flood
```

### Configuring SYN Flood Protection Options (CLI Procedure)

To set syn-flood parameters, use the following commands:

```
user@host# set security screen zone-syn-flood tcp syn-flood attack-threshold number
user@host# set security screen zone-syn-flood tcp syn-flood alarm-threshold number
user@host# set security screen zone-syn-flood tcp syn-flood source-threshold number
user@host# set security screen zone-syn-flood tcp syn-flood destination-threshold number
user@host# set security screen zone-syn-flood tcp syn-flood timeout number
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

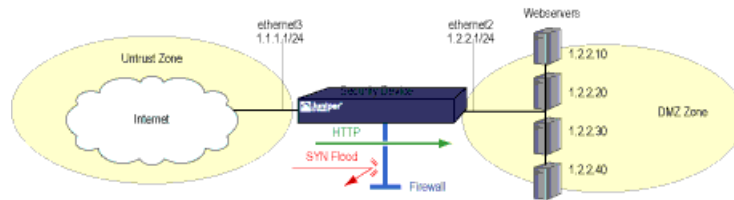
### Example: Enabling SYN Flood Protection for Webservers in the DMZ (CLI)

The following example shows you how to protect four web servers in the DMZ zone from SYN flood attacks originating in the external zone by enabling the SYN flood protection screen option for the external zone.



**NOTE:** We recommend that you augment the SYN flood protection that Junos OS provides with device-level SYN flood protection on each of the web servers. In this example, the web servers are running UNIX, which also provides some SYN flood defenses, such as adjusting the length of the connection request queue and changing the timeout period for incomplete connection requests.

Figure 77: Device-Level SYN Flood Protection



To configure the SYN flood protection parameters with appropriate values for your network, you must first establish a baseline of typical traffic flows. For one week, you run a sniffer on ethernet3—the interface bound to zone\_external—to monitor the number of new TCP connection requests arriving every second for the four web servers in the DMZ zone. Your analysis of the data accumulated from one week of monitoring produces the following statistics:

- Average number of new connection requests per server: 250 per second
- Average peak number of new connection requests per server: 500 per second



**NOTE:** A sniffer is a network-analyzing device that captures packets on the network segment to which you attach it. Most sniffers allow you to define filters to collect only the type of traffic that interests you. Later, you can view and evaluate the accumulated information. In this example, you want the sniffer to collect all TCP packets with the SYN flag set arriving at ethernet3 and destined for one of the four web servers in the DMZ. You might want to continue running the sniffer at regular intervals to see if there are traffic patterns based on the time of day, days of the week, the time of month, or the season of the year. For example, in some organizations, traffic might increase dramatically during a critical event. Significant changes probably warrant adjusting the various thresholds.

Based on this information, you set the following SYN flood protection parameters for zone\_external as shown in Table 70 on page 754.

Table 70: SYN Flood Protection Parameters

| Parameter        | Value                        | Reason for Each Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Attack Threshold | 625 packets per second (pps) | This is 25% higher than the average peak number of new connection requests per second per server, which is unusual for this network environment. When the number of SYN packets per second for any one of the four web servers exceeds this number, the device begins proxying new connection requests to that server. (In other words, beginning with the 626th SYN packet to the same destination address and port number in one second, the device begins proxying connection requests to that address and port number.) |
| Alarm Threshold  | 250 pps                      | When the device proxies 251 new connection requests in one second, it makes an alarm entry in the event log. By setting the alarm threshold somewhat higher than the attack threshold, you can avoid alarm entries for traffic spikes that only slightly exceed the attack threshold.                                                                                                                                                                                                                                       |

Table 70: SYN Flood Protection Parameters (*continued*)

| Parameter             | Value                        | Reason for Each Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Source Threshold      | 25 pps                       | <p>When you set a source threshold, the device tracks the source IP address of SYN packets, regardless of the destination address and port number. (Note that this source-based tracking is separate from the tracking of SYN packets based on destination address and destination port number that constitutes the basic SYN flood protection mechanism.)</p> <p>In the one week of monitoring activity, you observed that no more than 1/25 of new connection requests for all servers came from any one source within a one-second interval. Therefore, connection requests exceeding this threshold are unusual and provide sufficient cause for the device to execute its proxying mechanism. (25 pps is 1/25 of the attack threshold, which is 625 pps.)</p> <p>If the device tracks 25 SYN packets from the same source IP address, beginning with the 26th packet, it rejects all further SYN packets from that source for the remainder of that second and the next second as well.</p> |
| Destination Threshold | 0 pps                        | <p>When you set a destination threshold, the device runs a separate tracking of only the destination IP address, regardless of the destination port number. Because the four web servers only receive HTTP traffic (destination port 80)—no traffic to any other destination port number reaches them—setting a separate destination threshold offers no additional advantage.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Timeout               | 20 seconds                   | <p>The default value of 20 seconds is a reasonable length of time to hold incomplete connection requests.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Attack Threshold      | 625 packets per second (pps) | <p>This is 25% higher than the average peak number of new connection requests per second per server, which is unusual for this network environment. When the number of SYN packets per second for any one of the four web servers exceeds this number, the device begins proxying new connection requests to that server. (In other words, beginning with the 626th SYN packet to the same destination address and port number in one second, the device begins proxying connection requests to that address and port number.)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

To configure SYN flood protection parameters:

1. Set interfaces;

```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 1.2.2.1/24
user@host# set interfaces fe-1/0/0 unit 0 family inet address 1.1.1.1/24
user@host# set security zones security-zone zone_dmz interfaces ge-0/0/0.0
user@host# set security zones security-zone zone_external interfaces fe-1/0/0.0
```

2. Define addresses:

```
user@host# set security zones security-zone zone_dmz address-book address ws1
1.2.2.10/32
user@host# set security zones security-zone zone_dmz address-book address ws2
1.2.2.20/32
user@host# set security zones security-zone zone_dmz address-book address ws3
1.2.2.30/32
user@host# set security zones security-zone zone_dmz address-book address ws4
1.2.2.40/32
```

```
user@host# set security zones security-zone zone_dmz address-book address-set
web_servers address ws1
user@host# set security zones security-zone zone_dmz address-book address-set
web_servers address ws2
user@host# set security zones security-zone zone_dmz address-book address-set
web_servers address ws3
user@host# set security zones security-zone zone_dmz address-book address-set
web_servers address ws4
```

3. Configure the policy:

```
user@host# set security policies from-zone zone_external to-zone zone_dmz policy
id_1 match source-address any
user@host# set security policies from-zone zone_external to-zone zone_dmz policy
id_1 match destination-address web_servers
user@host# set security policies from-zone zone_external to-zone zone_dmz policy
id_1 match application junos-http
user@host# set security policies from-zone zone_external to-zone zone_dmz policy
id_1 then permit
```

4. Configure screen options:

```
user@host# set security screen ids-option zone_external-syn-flood tcp syn-flood
alarm-threshold 250
user@host# set security screen ids-option zone_external-syn-flood tcp syn-flood
attack-threshold 625
user@host# set security screen ids-option zone_external-syn-flood tcp syn-flood
source-threshold 25
user@host# set security screen ids-option zone_external-syn-flood tcp syn-flood
timeout 20
user@host# set security zones security-zone zone_external screen
zone_external-syn-flood
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## SYN Cookie Protection

- Understanding SYN Cookie Protection on page 756
- Example: Enabling SYN Cookie Protection (CLI) on page 758

### Understanding SYN Cookie Protection

SYN Cookie is a stateless SYN proxy mechanism you can use in conjunction with the defenses against a SYN flood attack.

As with traditional SYN proxying, SYN Cookie is activated when the SYN flood attack threshold is exceeded. However, because SYN Cookie is stateless, it does not set up a session or policy and route lookups upon receipt of a SYN segment, and it maintains no connection request queues. This dramatically reduces CPU and memory usage and is the primary advantage of using SYN Cookie over the traditional SYN proxying mechanism.

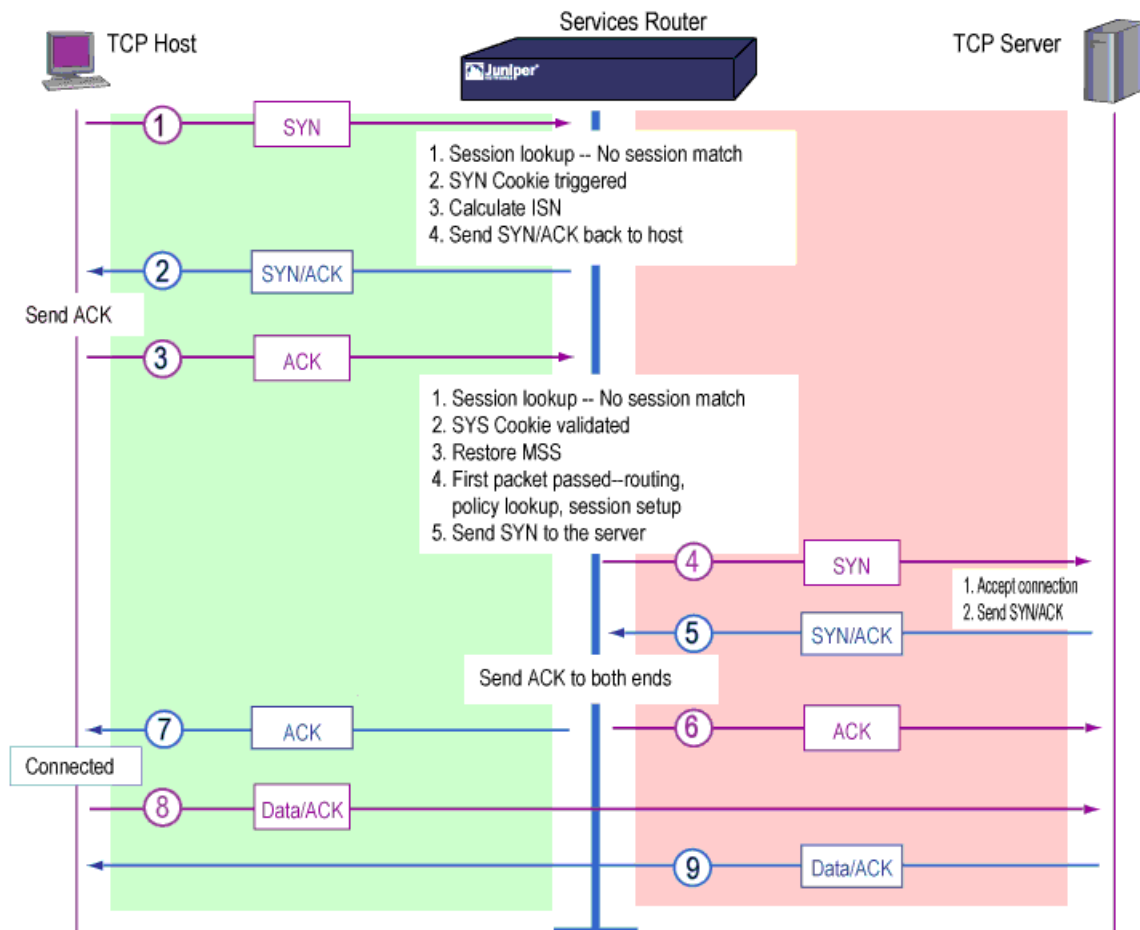
When SYN Cookie is enabled on Junos OS and becomes the TCP-negotiating proxy for the destination server, it replies to each incoming SYN segment with a SYN/ACK containing an encrypted cookie as its initial sequence number (ISN). The cookie is an MD5 hash of the original source address and port number, destination address and port

number, and ISN from the original SYN packet. After sending the cookie, Junos OS drops the original SYN packet and deletes the calculated cookie from memory. If there is no response to the packet containing the cookie, the attack is noted as an active SYN attack and is effectively stopped.

If the initiating host responds with a TCP packet containing the cookie +1 in the TCP ACK field, Junos OS extracts the cookie, subtracts 1 from the value, and recomputes the cookie to validate that it is a legitimate ACK. If it is legitimate, Junos OS starts the TCP proxy process by setting up a session and sending a SYN to the server containing the source information from the original SYN. When Junos OS receives a SYN/ACK from the server, it sends ACKs to the server and to the initiation host. At this point the connection is established and the host and server are able to communicate directly.

Figure 78 on page 757 shows how a connection is established between an initiating host and a server when SYN Cookie is active on Junos OS.

**Figure 78: Establishing a Connection with SYN Cookie Active**



**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Enabling SYN Cookie Protection (CLI)

The following example shows how to set the SYN flood attack threshold.



NOTE: The SYN Cookie feature can only detect and protect against spoofed SYN flood attacks, thus minimizing the negative impact to hosts that are secured by Junos OS. If an attacker is using a legitimate IP source address, rather than a spoofed IP source, then the SYN-Cookie mechanism does not stop the attack.

```
user@host# set security screen ids-option external-syn-flood tcp syn-flood timeout 20
user@host# set security zones security-zone external screen external-syn-flood
user@host# set security flow syn-flood-protection-mode syn-cookie
```

**Related Topics**   • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## ICMP Flood Protection

- Understanding ICMP Flood Attacks on page 758
- Example: Enabling ICMP Flood Protection (CLI) on page 759

### Understanding ICMP Flood Attacks

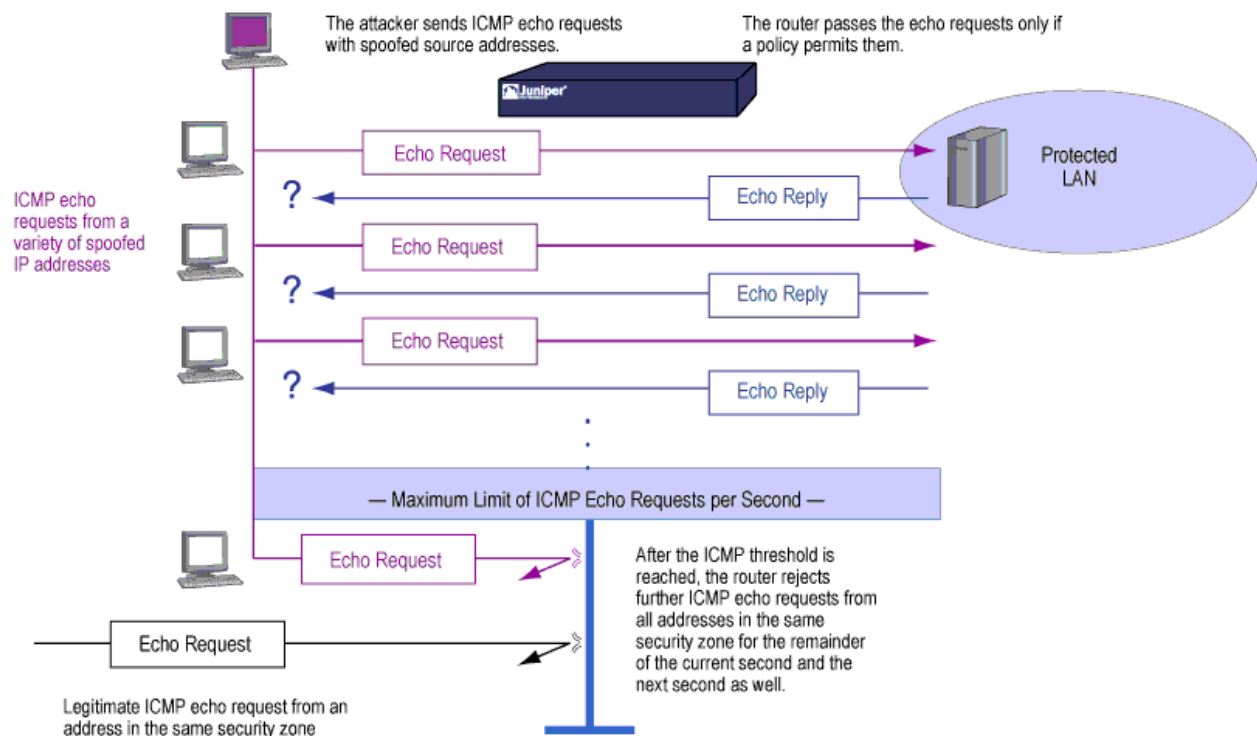
An ICMP flood typically occurs when ICMP echo requests overload the victim with so many requests that the victim expends all its resources responding until it can no longer process valid network traffic.

When enabling the ICMP flood protection feature, you can set a threshold that, once exceeded, invokes the ICMP flood attack protection feature. (The default threshold value is 1000 packets per second.) If the threshold is exceeded, Junos OS ignores further ICMP echo requests for the remainder of that second plus the next second as well. See Figure 79 on page 759.



NOTE: An ICMP flood can consist of any type of ICMP message. Therefore, Junos OS monitors all ICMP message types, not just echo requests.

Figure 79: ICMP Flooding



**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

#### Example: Enabling ICMP Flood Protection (CLI)

The following example shows you how to enable ICMP flood protection. (The value unit is ICMP packets per second, or pps. The default value is 1000 pps.) In the example, the specified zone is where a flood might originate.

```
user@host# set security screen ids-option 1000-icmp-flood icmp flood threshold 1000
user@host# set security zones security-zone zone screen 1000-icmp-flood
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## UDP Flood Attacks

- Understanding UDP Flood Attacks on page 759
- Example: Enabling UDP Flood Protection (CLI) on page 760

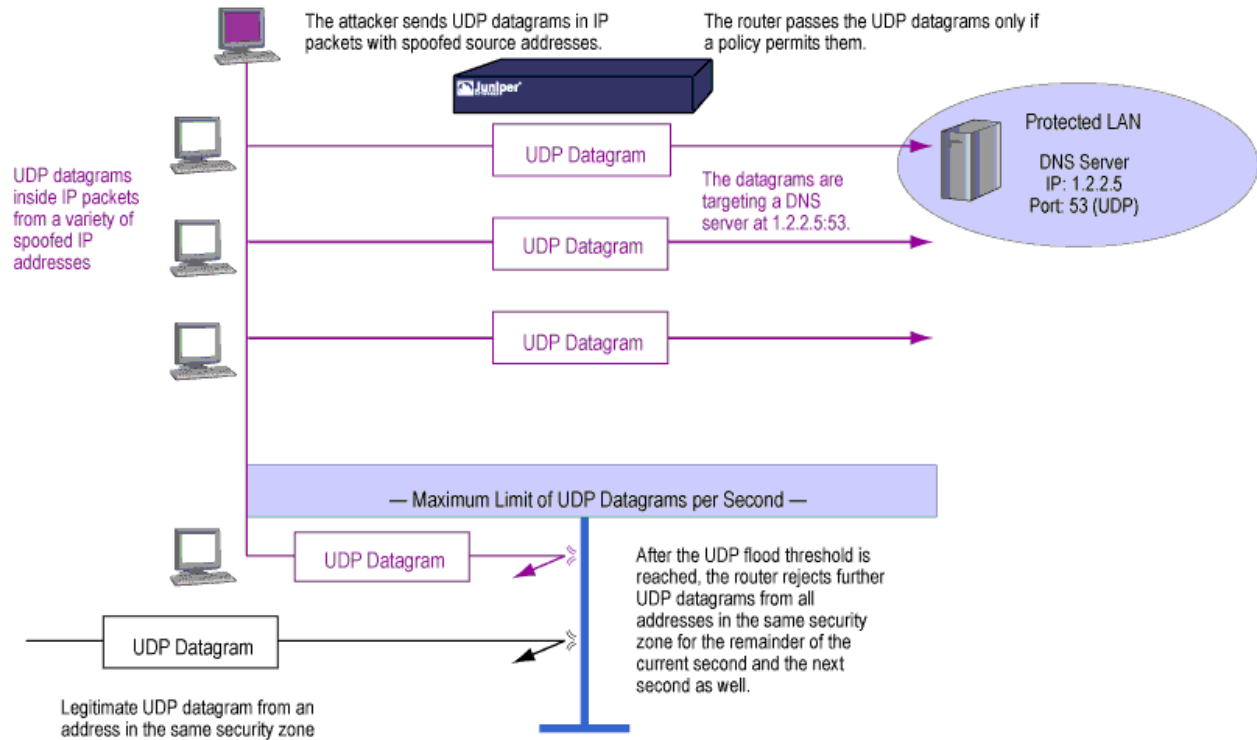
### Understanding UDP Flood Attacks

Similar to an ICMP flood, a UDP flood occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that the victim can no longer handle valid connections.

After enabling the UDP flood protection feature, you can set a threshold that, once exceeded, invokes the UDP flood attack protection feature. (The default threshold value is 1000 packets per second, or pps.) If the number of UDP datagrams from one or more

sources to a single destination and UDP port exceeds this threshold, Junos OS ignores further UDP datagrams to that destination and port for the remainder of that second plus the next second as well. See Figure 80 on page 760.

**Figure 80: UDP Flooding**



**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Enabling UDP Flood Protection (CLI)

The following example shows you how to enable UDP flood protection. (The value unit is UDP packets per second, or pps. The default value is 1000 pps.) In the example, the specified zone is where a flood might originate.

```
user@host# set security zones security-zone external screen external-udp-flood
user@host# set security screen ids-option 1000-udp-flood udp flood threshold 1000
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Land Attacks

- Understanding Land Attacks on page 760
- Example: Protecting Against a Land Attack (CLI) on page 761

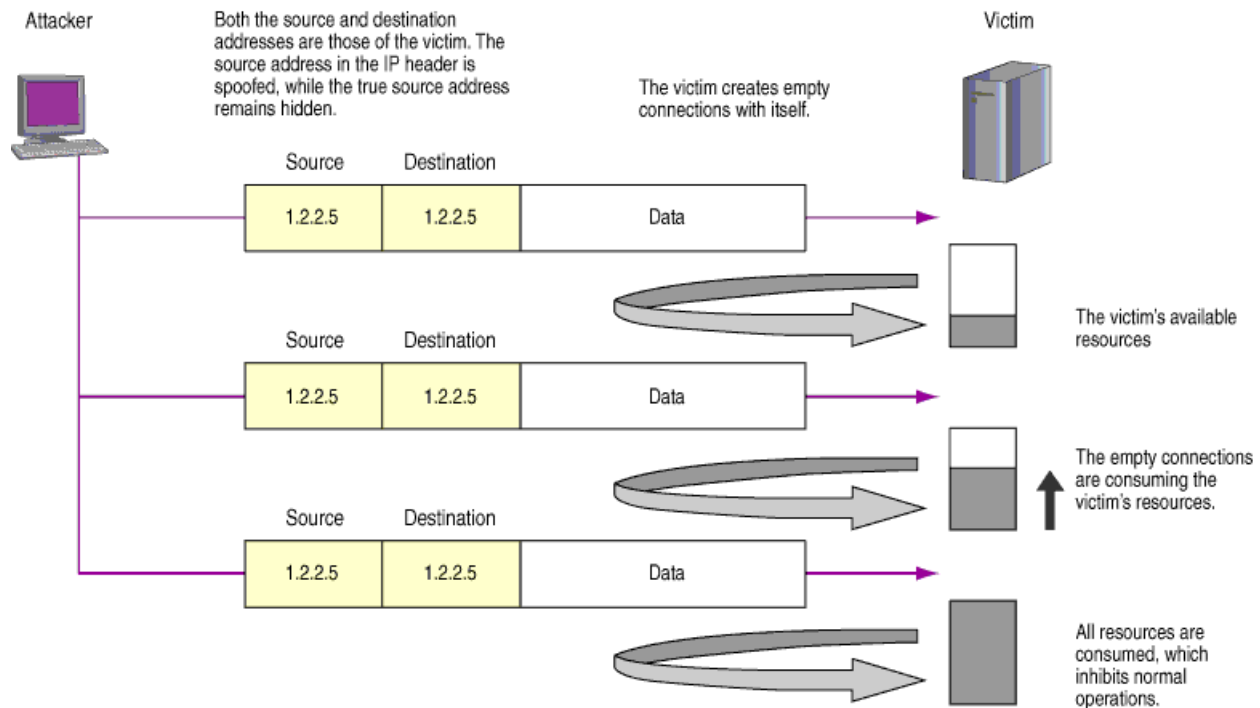
### Understanding Land Attacks

Combining a SYN attack with IP spoofing, a land attack occurs when an attacker sends spoofed SYN packets containing the IP address of the victim as both the destination and the source IP address.



The receiving system responds by sending the SYN-ACK packet to itself, creating an empty connection that lasts until the idle timeout value is reached. Flooding a system with such empty connections can overwhelm the system, causing a denial of service (DoS). See Figure 81 on page 761.

**Figure 81: Land Attack**



When you enable the screen option to block land attacks, Junos OS combines elements of the SYN flood defense and IP spoofing protection to detect and block any attempts of this nature.

#### Example: Protecting Against a Land Attack (CLI)

The following example shows you how to enable protection against a land attack.

```
user@host# set security screen land tcp land
user@host# set security zones security-zone zone screen land
```

## OS-Specific DoS Attacks

- OS-Specific DoS Attacks Overview on page 761
- Ping of Death Attacks on page 762
- Teardrop Attacks on page 763
- WinNuke Attacks on page 764

## OS-Specific DoS Attacks Overview

If an attacker not only identifies the IP address and responsive port numbers of an active host but also its operating system (OS), instead of resorting to brute-force attacks, the

attacker can launch more elegant attacks that can produce one-packet or two-packet “kills.”

OS-specific denial-of-service (DoS) attacks, including ping of death attacks, teardrop attacks, and WinNuke attacks, can cripple a system with minimal effort. If Junos OS is protecting hosts susceptible to these attacks, you can configure Junos OS to detect these attacks and block them before they reach their target.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Ping of Death Attacks

- Understanding Ping of Death Attacks on page 762
- Example: Protecting Against a Ping of Death Attack (CLI) on page 763

### Understanding Ping of Death Attacks

OS-specific DoS attacks, such as ping of death attacks, can cripple a system with minimal effort.

The maximum allowable IP packet size is 65,535 bytes, including the packet header, which is typically 20 bytes. An ICMP echo request is an IP packet with a pseudo header, which is 8 bytes. Therefore, the maximum allowable size of the data area of an ICMP echo request is 65,507 bytes ( $65,535 - 20 - 8 = 65,507$ ).

However, many ping implementations allow the user to specify a packet size larger than 65,507 bytes. A grossly oversized ICMP packet can trigger a range of adverse system reactions such as denial of service (DoS), crashing, freezing, and rebooting.

When you enable the ping of death screen option, Junos OS detects and rejects such oversized and irregular packet sizes even when the attacker hides the total packet size by fragmenting it. See Figure 82 on page 762.



**NOTE:** For information about IP specifications, see RFC 791, *Internet Protocol*. For information about ICMP specifications, see RFC 792, *Internet Control Message Protocol*. For information about ping of death attacks, see <http://www.insecure.org/sploits/ping-o-death.html>.

**Figure 82: Ping of Death**



The size of this packet is 65,538 bytes. It exceeds the size limit prescribed by RFC 791, *Internet Protocol*, which is 65,535 bytes. As the packet is transmitted, it becomes broken into numerous fragments. The reassembly process might cause the receiving system to crash.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

### Example: Protecting Against a Ping of Death Attack (CLI)

The following example shows you how to enable protection against a ping of death attack. In the example, the specified zone is where the attack originates.

```
user@host# set security screen ids-option ping-death icmp ping-death
user@host# set security zones security-zone zone screen ping-death
```

**Related Topics**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## Teardrop Attacks

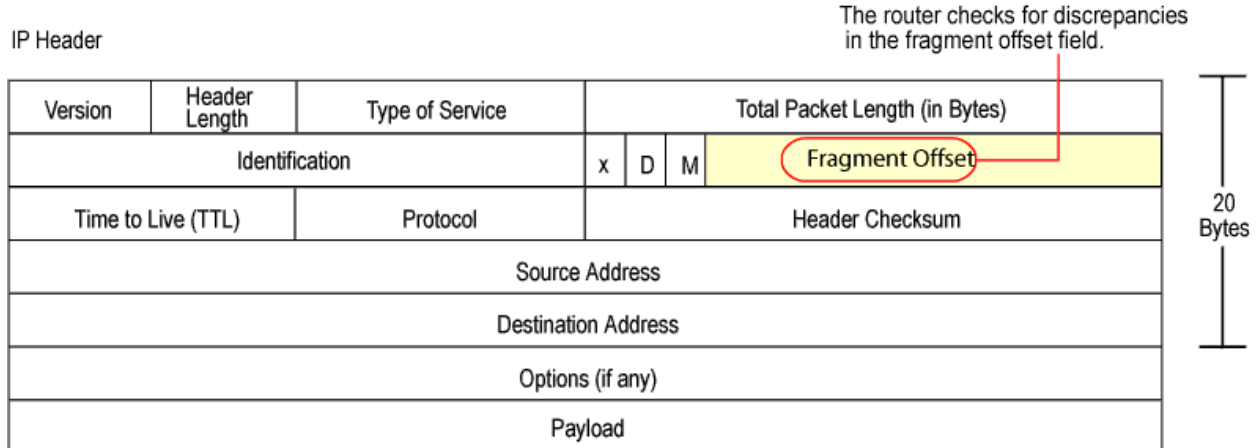
- Understanding Teardrop Attacks on page 763
- Example: Protecting Against a Teardrop Attack (CLI) on page 764

### Understanding Teardrop Attacks

OS-specific denial-of-service (DoS) attacks, such as teardrop attacks, can cripple a system with minimal effort.

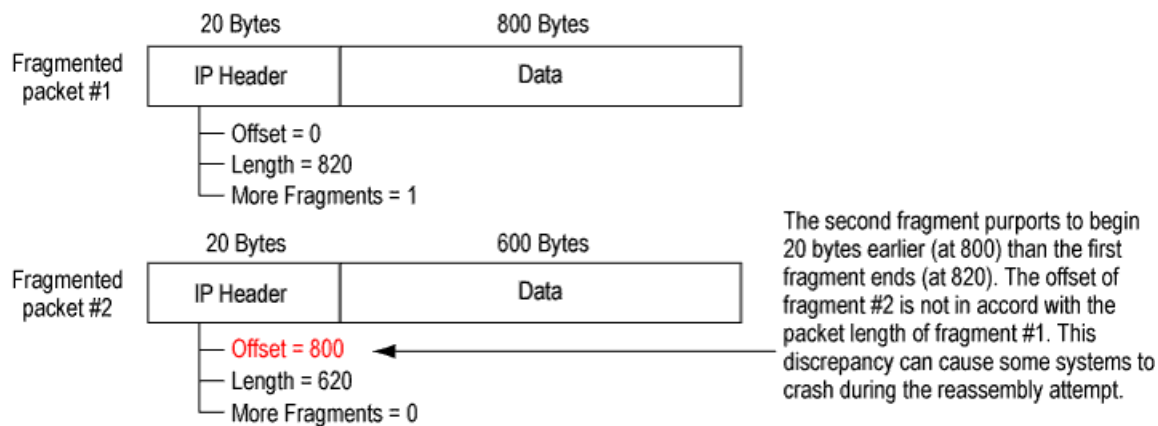
Teardrop attacks exploit the reassembly of fragmented IP packets. In the IP header, one of the fields is the fragment offset field, which indicates the starting position, or offset, of the data contained in a fragmented packet relative to the data of the original unfragmented packet. See Figure 83 on page 763.

Figure 83: Teardrop Attacks



When the sum of the offset and size of one fragmented packet differ from that of the next fragmented packet, the packets overlap, and the server attempting to reassemble the packet can crash, especially if it is running an older OS that has this vulnerability. See Figure 84 on page 764.

Figure 84: Fragment Discrepancy



After you enable the teardrop attack screen option, whenever Junos OS detects this discrepancy in a fragmented packet, it drops it.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

#### Example: Protecting Against a Teardrop Attack (CLI)

The following example shows you how to enable protection against a teardrop attack. In the example, the specified zone is where the attack originates.

```
user@host# set security screen ids-option tear-drop ip tear-drop
user@host# set security zones security-zone zone screen tear-drop
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

## WinNuke Attacks

- Understanding WinNuke Attacks on page 764
- Example: Protecting Against a WinNuke Attack (CLI) on page 765

### Understanding WinNuke Attacks

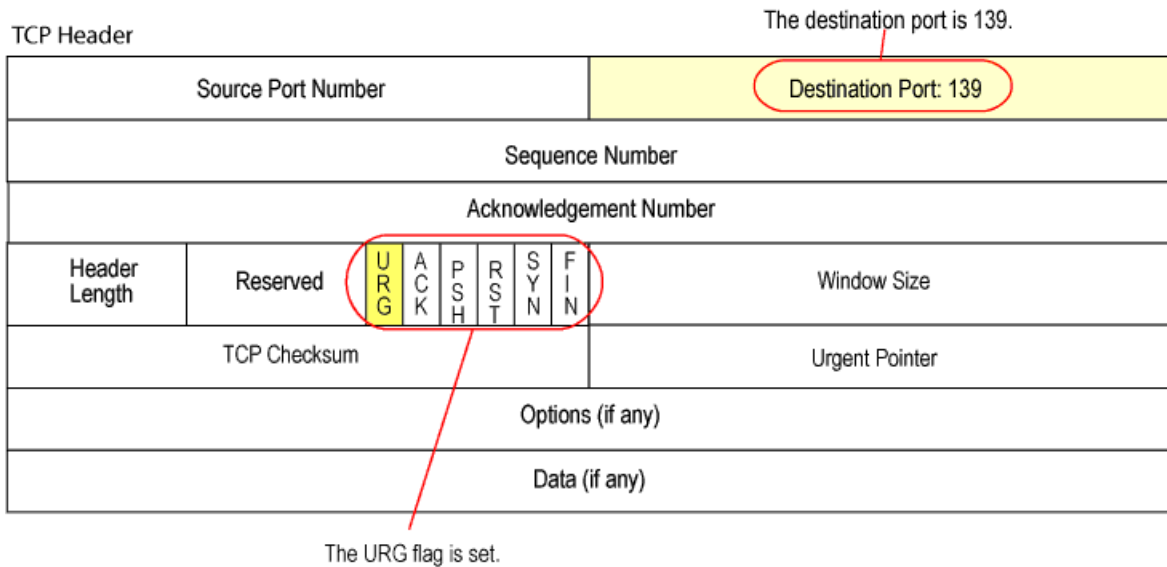
OS-specific denial-of-service (DoS) attacks, such as WinNuke attacks, can cripple a system with minimal effort.

WinNuke is a DoS attack targeting any computer on the Internet running Windows. The attacker sends a TCP segment—usually to NetBIOS port 139 with the urgent (URG) flag set—to a host with an established connection (see Figure 85 on page 765). This introduces a NetBIOS fragment overlap, which causes many machines running Windows to crash. After the attacked machine is rebooted, the following message appears, indicating that an attack has occurred:

```
An exception OE has occurred at 0028:[address] in VxD MSTCP(01) +
000041AE. This was called from 0028:[address] in VxD NDIS(01) +
00008660. It may be possible to continue normally.
Press any key to attempt to continue.
Press CTRL+ALT+DEL to restart your computer. You will lose any unsaved information in
all applications.
```

Press any key to continue.

Figure 85: WinNuke Attack Indicators



If you enable the WinNuke attack defense screen option, Junos OS scans any incoming Microsoft NetBIOS session service (port 139) packets. If Junos OS observes that the URG flag is set in one of those packets, it unsets the URG flag, clears the URG pointer, forwards the modified packet, and makes an entry in the event log noting that it has blocked an attempted WinNuke attack.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

#### Example: Protecting Against a WinNuke Attack (CLI)

The following example shows you how to enable protection against a WinNuke attack. In the example, the specified zone is where the attack originates.

```
user@host# set security screen winnuke tcp winnuke
user@host# set security zones security-zone zone screen winnuke
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*



## PART 10

# Application Identification

- Junos OS Application Identification on page 769
- AppTrack Application Tracking on page 789





# Junos OS Application Identification

- Understanding Junos OS Application Identification Services on page 769
- Application Identification Application Package on page 770
- Disabling Junos OS Application Identification (CLI Procedure) on page 775
- Junos OS Application Identification for Nested Applications on page 776
- Junos OS Application Identification Custom Application Signature Definitions on page 777
- Application System Cache on page 782
- Memory and Session Limits on page 786

## Understanding Junos OS Application Identification Services

---

Juniper Networks provides predefined application signatures that detect Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) applications running on nonstandard ports. You can also create custom application and nested application definitions to identify applications that are not part of the predefined database. Identifying these applications allows Intrusion Detection and Prevention (IDP) to apply appropriate attack objects to applications running on nonstandard ports. It also improves performance by narrowing the scope of attack signatures for applications without decoders. The information collected by application identification can also be used by AppTrack to create detailed reports on the applications passing through the device.

The application definitions identify an application by matching patterns in the first few packets of a session. The application identification module matches patterns for both client-to-server and server-to-client sessions.

Application identification is enabled by default and is automatically turned on when you configure the default application in an IDP or an AppTrack policy. However, when you specify an application in the policy rule, application identification is disabled and attack objects are applied based on the specified application. This specific application configuration overwrites the automatic identification process. For instructions on specifying applications in policy rules, see “Example: Configuring IDP Applications and Services” on page 494.

For information on IDP application identification, see “Understanding IDP Application Identification” on page 549.

For information on AppTrack, see “Understanding AppTrack” on page 789.



NOTE: The Junos OS application identification application signature package update is a separately licensed subscription service. You must install the application identification application signature update license key on your device to download and install the signature database updates provided by Juniper Networks. When your license key expires, you can continue to use the locally stored application package contents. For license details, see the *Junos OS Administration Guide for Security Devices*

---

**Related Topics**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Junos OS Application Identification Application Package on page 770
- Understanding Junos OS Application Identification Custom Application Definitions on page 777
- IDP Policies Overview on page 463
- Understanding IDP Service and Application Bindings by Attack Objects on page 550

---

## Application Identification Application Package

---

- Understanding Junos OS Application Identification Application Package on page 770
- Updating Junos OS Application Identification Extracted Application Package Overview on page 771
- Updating Junos OS Application Identification Extracted Application Package Manually Overview on page 772
- Example: Updating Junos OS Application Identification Extracted Application Package Manually (CLI) on page 772
- Example: Updating Junos OS Application Identification Extracted Application Package Automatically (CLI) on page 773
- Example: Verifying Junos OS Application Identification Extracted Application Package on page 774

## Understanding Junos OS Application Identification Application Package

Juniper Networks regularly updates the predefined application identification application package database that is part of the IDP signature database and makes it available on the Juniper Networks website. This package includes a list of known application objects that can be used in Intrusion Detection and Prevention (IDP) and AppTrack to match traffic. You need to download the application package before configuring application identification or AppTrack.

The application package contains application objects such as ftp and DNS as well as nested applications such as Facebook, Kazaa, and many instant messenger programs. The application database is visible in the configuration, and custom application definitions can be created. For information on custom definitions, see “Understanding Junos OS Application Identification Custom Application Definitions” on page 777. If you do not have IDP enabled and will use application identification with AVT, you will run the following command: **request services application-identification download**. This command will

extract and install the application portion of the IDP signature database to your configuration. If you have IDP enabled and will use application identification, you will continue to run the IDP signature database download. To download the IDP signature database run the following command: **request security idp security-package download**. The application package download can be performed manually or automatically.



**NOTE:** If you have an IDP-enabled device and will use application identification, we recommend that you only download the IDP signature database. This will avoid having two versions of the application database, which may become out of sync. For information on the IDP signature database download that contains its own application database, see “Understanding the IDP Signature Database” on page 535.



**NOTE:** The Junos OS application identification application signature package update is a separately licensed subscription service. You must install the application identification application signature update license key on your device to download and install the signature database updates provided by Juniper Networks. When your license key expires, you can continue to use the locally stored application package contents. For license details, see the *Junos OS Administration Guide for Security Devices*

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Junos OS Application Identification Services on page 769
- Updating Junos OS Application Identification Extracted Application Package Overview on page 771
- Example: Updating Junos OS Application Identification Extracted Application Package Manually (CLI) on page 772
- Example: Updating Junos OS Application Identification Extracted Application Package Automatically (CLI) on page 773

## Updating Junos OS Application Identification Extracted Application Package Overview

Juniper Networks regularly updates the predefined application identification application package database and makes it available on the Juniper Networks website. The application package download can be performed manually or automatically, and the download command handles the download and installation of the application package. If you have an IDP-enabled device and will use application identification, we recommend that you only download the IDP signature database. This will avoid having two versions of the application database, which may become out of sync. For information on the IDP signature database download that contains its own application database, see “Understanding the IDP Signature Database” on page 535.



NOTE: Uninstalling the application package will not remove any custom application or nested application definitions that you have created. All predefined Juniper applications have the prefix “junos”, so make sure you do not use “junos” for your custom definition names.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Junos OS Application Identification Services on page 769
  - Example: Updating Junos OS Application Identification Extracted Application Package Manually (CLI) on page 772
  - 
  - Example: Verifying Junos OS Application Identification Extracted Application Package on page 774

## Updating Junos OS Application Identification Extracted Application Package Manually Overview

Juniper Networks regularly updates the predefined application identification application package database and makes it available on the Juniper Networks website. The extracted package can be download manually in order to watch the install process, change the download URL, or import custom application or nested application definition files.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Updating Junos OS Application Identification Extracted Application Package Manually Overview on page 772
  - Example: Updating Junos OS Application Identification Extracted Application Package Manually (CLI) on page 772
  - Example: Updating Junos OS Application Identification Extracted Application Package Automatically (CLI) on page 773
  - Example: Verifying Junos OS Application Identification Extracted Application Package on page 774

## Example: Updating Junos OS Application Identification Extracted Application Package Manually (CLI)

Before you begin, configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

Juniper Networks regularly updates the predefined application identification application package database and makes it available on the Juniper Networks website. This package includes a list of known application objects that can be used in Intrusion Detection and Prevention (IDP) policies and AppTrack to match traffic.

The configuration instructions in this topic describes how to download the application identification application package and create a policy, and specify the new policy as the active policy. The download process will also install the application package.

1. To manually download and update the application package:

```
user@host> request services application-identification download
```

To download a specific version of the application package:

```
user@host> request services application-identification download version
version-number
```

To change the download URL for the application package from configuration mode:

```
[edit]
user@host# set services application-identification download url URL or File Path
```



**NOTE:** If you change the download URL and you want to keep that change, make sure you commit.

To uninstall the application package:

```
user@host> request services application-identification uninstall
```

2. To check the current version of the application package:

```
show services application-identification version
```

3. The application package will now be part of your configuration. From configuration mode in the CLI, enter the **show services application-identification** command to verify the configuration. For more information, see the *Junos OS CLI Reference*.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Junos OS Application Identification Application Package on page 770
- Example: Updating Junos OS Application Identification Extracted Application Package Automatically (CLI) on page 773
- Example: Verifying Junos OS Application Identification Extracted Application Package on page 774

### Example: Updating Junos OS Application Identification Extracted Application Package Automatically (CLI)

Juniper Networks regularly updates the predefined application identification application package database and makes it available on the Juniper Networks website.

The configuration instructions in this topic describe how to download the application package automatically.

1. Specify the time and interval for download. Automatic download will be activated as soon as you set a start time. If no interval is specified, the default of 24 hours will be set. The interval range is 6 through 720 hours.

The following statement sets the interval to every 48 hours with a start time of 11:59 pm on December 10:

```
user@host# set services application-identification download automatic interval 48
start-time 12-10.23:59
```

2. If you are finished configuring the device, commit the configuration.
3. From configuration mode in the CLI, enter the **show services application identification** command to verify the configuration. For more information, see the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Junos OS Application Identification Application Package on page 770
  - Updating Junos OS Application Identification Extracted Application Package Manually Overview on page 772
  - Example: Verifying Junos OS Application Identification Extracted Application Package on page 774

### Example: Verifying Junos OS Application Identification Extracted Application Package

Juniper Networks regularly updates the predefined application identification application package database and makes it available on the Juniper Networks website, so it is important that you have the most recent version.

When you download the application package from the IDP signature database, you will see a status message after you enter the download command. Example, on successful download, you will see the following message (where xxxx is the package version number):

application package xxxx is downloaded successfully

The syslog will also show the result of the download.

To view the contents of the application package that is inserted into the configuration after successful download:

```
show services application identification
```

The output that follows shows the first entry in the application package database, which is the predefined AIM application:

```
application junos:AIM {
 type AIM;
 index 61;
 port-mapping {
 port-range {
 tcp 5190;
 }
 }
 signature {
 port-range {
 tcp 0-65535;
 }
 client-to-server {
 dfa-pattern "(*\01[\^\07]*\00.*|CONNECT login\.oscar\.aol\.com)\.*";
 }
 server-to-client {
 dfa-pattern "(*\01|HTTP/1\.[01] 200 Connection established\x0d 0a
0d 0a\x)\.*";
 }
 min-data 10;
 }
}
```

```
 order 9;
 }
}
```

To check the version of the current application package from configuration mode (the version information will be the first line item):

**show services application-identification**

To check the version from operational mode:

**show services application-identification version**

You will see the following output if package version 1608 is installed successfully:

```
Application package version: 1608
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Junos OS Application Identification Application Package on page 770
  - Example: Updating Junos OS Application Identification Extracted Application Package Automatically (CLI) on page 773
  - Updating Junos OS Application Identification Extracted Application Package Manually Overview on page 772
  - Example: Verifying Junos OS Application Identification Extracted Application Package on page 774

---

## Disabling Junos OS Application Identification (CLI Procedure)

Application identification is enabled by default. You can disable application identification with the CLI.

To disable application identification:

```
user@host# set services application-identification no-application-identification
```

If you want to reenable application identification, delete the configuration statement that specifies disabling of application identification:

```
user@host# delete services application-identification no-application-identification
```

If you are finished configuring the device, commit the configuration.

To verify the configuration, enter the **show services application-identification** command. For more information, see the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Junos OS Application Identification Services on page 769

## Junos OS Application Identification for Nested Applications

---

- Understanding Junos OS Application Identification for Nested Applications on page 776
- Activating Junos OS Application Identification for Nested Applications (CLI Procedure) on page 776

### Understanding Junos OS Application Identification for Nested Applications

With the greater use of application protocol encapsulation, the need arises to support the identification of multiple different applications running on the same Layer 7 protocols. For example, applications such as Facebook and Yahoo Messenger can both run over HTTP, but there is a need to identify them as two different applications running on the same Layer 7 protocol. In order to do this, the current application identification layer is split into two layers: Layer 7 nested applications and Layer 7 protocols.

The included predefined application signatures have been created to detect the Layer 7 nested applications whereas the existing Layer 7 protocol signatures, such as FTP and HTTP, still function in the same manner. These predefined application signatures can be used in attack objects.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Junos OS Application Identification Application Package on page 770
  - Activating Junos OS Application Identification for Nested Applications (CLI Procedure) on page 776
  - Understanding Junos OS Application Identification Services on page 769

### Activating Junos OS Application Identification for Nested Applications (CLI Procedure)

Application identification for nested applications is turned on by default. You can manually turn it off by using the CLI.

To disable nested application identification:

1. `user@host# set services application-identification nested-application-settings no-nested-application`
2. If you want to reenable nested application identification, delete the configuration statement:  
`user@host# delete services application-identification nested-application-settings no-nested-application`
3. If you are finished configuring the device, commit the configuration.
4. To verify the configuration, enter the `show services application-identification nested-applications` command. For more information, see the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Junos OS Application Identification for Nested Applications on page 776



- Understanding Junos OS Application Identification Application Package on page 770
- Understanding Junos OS Application Identification Services on page 769

## Junos OS Application Identification Custom Application Signature Definitions

---

- Understanding Junos OS Application Identification Custom Application Definitions on page 777
- Example: Configuring Junos OS Application Identification Custom Application Definitions (CLI) on page 777
- Example: Configuring Junos OS Application Identification Custom Nested Application Definitions (CLI) on page 780

## Understanding Junos OS Application Identification Custom Application Definitions

Application identification supports user-defined custom application definitions for applications and nested applications. With custom application definitions you can create definitions that will detect applications that are not part of the predefined application package. Both predefined and custom application definitions are located in the [services application-identification application] hierarchy. The predefined and custom applications for nested application definitions are located in the [services application-identification nested-application] hierarchy. When you perform an update or uninstall the application package, custom applications will not be modified or removed.

When you create custom application or nested application definitions, make sure your entries are unique to entries in the predefined application database. All predefined definitions provided by Juniper have the prefix “junos” in the definition name, for example junos:ftp, junos:facebook, so do not use that prefix when naming your custom definitions. Also, custom application definitions and custom nested application definitions share the same index pool, so the index entries must be unique among all application and nested application custom definitions. Once you download the application definition package, you can view definitions by running the **show services application-identification** command. You can use the predefined definitions as a base for creating your custom definitions; however, make sure your application name does not start with junos and that the index number of each definition is unique.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Configuring Junos OS Application Identification Custom Application Definitions (CLI) on page 777
  - Understanding Junos OS Application Identification Application Package on page 770
  - Understanding Junos OS Application Identification Services on page 769

## Example: Configuring Junos OS Application Identification Custom Application Definitions (CLI)

Application identification supports custom application definitions to detect applications as they pass through the device. When you configure custom definitions, make sure your definitions are unique.

Table 71 on page 778 shows the comparison between custom and predefined configuration parameters for applications. These differences will ensure that custom application definitions are unique to the predefined definitions so they are not deleted when updating or deleting the predefined application package.

**Table 71: Custom Application Definitions and Predefined Definitions Comparison**

| Predefined                                                      | Custom                                                                    |
|-----------------------------------------------------------------|---------------------------------------------------------------------------|
| index range: 1 through 32767                                    | index range: 32768 through 65534                                          |
| name prefix junos                                               | name prefix is user defined (junos is reserved for predefined signatures) |
| order field unique for all applications and nested applications | order field unique for all applications and nested applications           |

Table 72 on page 778 shows the available attributes for creating a custom application definition. The hierarchy level is [edit services application-identification application *application-name*].

**Table 72: Custom Application Definition Attributes**

| Attribute            | Description                                                                                                                                                                                                                                                                                              |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| application-name     | Name of the custom application definition. Must be a unique name with a maximum length of 32 characters. (Required)                                                                                                                                                                                      |
| disable              | Do not match traffic for this application. Default is off.                                                                                                                                                                                                                                               |
| index                | A number that is a one-to-one mapping to the application name that is used to ensure that each signature definition is unique. The index range for predefined applications is 1 through 32767. The index range for custom applications and custom nested applications is 32768 through 65534. (Required) |
| Signature Attributes |                                                                                                                                                                                                                                                                                                          |
| signature            | Define the application signature attributes for pattern matching. (Required)                                                                                                                                                                                                                             |
| client-to-server     | Defines the attributes for traffic in the client-to-server direction.<br><br>dfa-pattern: Maximum length is 1023. (Optional)<br><br>regex: Enter a regular expression that should be matched for client to server traffic.                                                                               |
| disable              | Toggle on means a signature method is not used to identify this application. Default is off.                                                                                                                                                                                                             |
| min-data             | The minimum number of bytes or packets to apply to the dfa-pattern. Default is 10, range is 4 through 1024.                                                                                                                                                                                              |
| order                | When there are multiple patterns matched for the same session, the lowest order number takes the highest priority. Must be unique. (Required)                                                                                                                                                            |

Table 72: Custom Application Definition Attributes (*continued*)

| Attribute        | Description                                                                                                                                                                                                                                                         |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| port-range       | Default ranges: TCP/0 through 65535, UDP/0 through 65535. (Optional)                                                                                                                                                                                                |
| server-to-client | <p>Defines the attributes for traffic in the client-to-server direction.</p> <p>dfa-pattern: Default ranges: TCP/0 through 65535, UDP/0 through 65535. (Optional)</p> <p>regex: Enter a regular expression that should be matched for server-to-client traffic.</p> |

The following example identifies an application named “my-app” operating over the HTTP protocol on TCP port 6400 with a signature port range of TCP 0–65535.

1. Set the application name you will use in your policy for your custom application.

```
[edit services application identification]
user@host# set application my-app
```

2. Set the application type.

```
[edit services application identification]
user@host# set application my-app type HTTP
```

3. Set the index number.

```
[edit services application identification]
user@host# set application my-app index 33000
```

4. Set the signature information by starting with the signature port range.

```
[edit services application identification]
user@host# set application my-app signature port-range tcp 0–65535.
```

5. Set the signature client-to-server dfa-pattern.

```
[edit services application identification]
user@host# set application my-app signature client-to-server dfa-pattern
\xff\x[\xfa-\xff].*
```

6. Set the signature server-to-client dfa-pattern.

```
[edit services application identification]
user@host# set application my-app signature server-to-client dfa-pattern
\xff\x[\xfa-\xff].*
```

7. Set the signature min data value

```
[edit services application identification]
user@host# set application my-app signature min-data 2
```

8. Set the signature order.

```
[edit services application identification]
user@host# set application my-app signature order 102
```

Now that your custom definition has been defined, you can use it in your policy. For information on policies, see “Example: Configuring IDP Policies for Application Identification (CLI)” on page 551.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Junos OS Application Identification Custom Application Definitions on page 777
  - Understanding Junos OS Application Identification Application Package on page 770
  - Understanding Junos OS Application Identification Services on page 769

### Example: Configuring Junos OS Application Identification Custom Nested Application Definitions (CLI)

Application identification supports custom nested application definitions to detect nested applications as they pass through the device. When you configure custom definitions, make sure your definitions are unique.

Table 73 on page 780 shows the comparison between custom and predefined configuration parameters for nested applications. These differences will ensure that custom nested application definitions are unique to the predefined definitions so they are not deleted when you update or delete the predefined application package.

**Table 73: Custom Nested Application Definitions and Predefined Definitions**

| Predefined                                                       | Custom                                                           |
|------------------------------------------------------------------|------------------------------------------------------------------|
| index range: 1 through 32767                                     | index range: 32768 through 65534                                 |
| name prefix: junos                                               | name prefix: not unique and must not be junos                    |
| order field: unique for all applications and nested applications | order field: unique for all applications and nested applications |

Table 74 on page 780 shows the available attributes for creating a custom nested application definition. The hierarchy level is [edit services application-identification nested-application *nested-application-name*].

**Table 74: Custom Nested Application Definition Attributes**

| Attribute               | Description                                                                                                                                                                               |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nested-application-name | Name of the custom nested application definition. Must be a unique name with a maximum length of 32 characters. (Required)                                                                |
| index                   | A number that is a one-to-one mapping to the nested application name. Must be unique with a maximum length of 32 bits. 1 through 1023 is reserved for predefined applications. (Required) |
| protocol                | The protocol that will be monitored to identify nested applications. HTTP is supported.                                                                                                   |

Table 74: Custom Nested Application Definition Attributes (*continued*)

| Attribute                         | Description                                                                                                                                                                                                     |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Signature Attributes              |                                                                                                                                                                                                                 |
| <code>signature name</code>       | Name of the custom nested application signature definition. Must be a unique name with a maximum length of 32 characters. (Required)                                                                            |
| <code>chain-order</code>          | Signatures can contain multiple members. If chain-order is on, those members are read in order. The default for this option is no chain order. If a signature only contains one member, this option is ignored. |
| <code>maximum-transactions</code> | The maximum number of transactions that should occur before a match is made.                                                                                                                                    |
| <code>member name</code>          | Defines a member name for a custom nested application signature definition. Custom definitions can contain multiple members that define attributes for an application. (Member name range is m01 through m16)   |
| <code>context</code>              | Defines a service specific context, such as http-url.                                                                                                                                                           |
| <code>direction</code>            | The connection direction of the packets to apply pattern matching. The options are any, client-to-server, or server-to-client.                                                                                  |
| <code>pattern</code>              | Define the dfa-pattern to match in the context.                                                                                                                                                                 |
| <code>order</code>                | When there are multiple patterns matched for the same session, the lowest order number takes the highest priority. Must be unique. (Required)                                                                   |

The following example identifies an application named “my-nested-app” for the nested application called Social-Website operating over HTTP.

To create a custom nested application definition:

1. Set the application name you will use in your policy for your custom application.

```
[edit services application identification]
user@host# set nested-application my-nested-app
```

2. Set the application type.

```
[edit services application identification]
user@host# set nested-application my-nested-app type HTTP
```

3. Set the index number.

```
[edit services application identification]
user@host# set nested-application my-app index 34000
```

4. Set the signature information by starting with the signature name my-nested-app-sig:Social-Website.

```
[edit services application identification]
user@host# set nested-application my-nested-app signature
my-nested-app-sig:Social-Website
```

5. Create a member named m01 for the signature that defines the application attributes. (member name range is m01 through m16)

```
[edit services application identification]
user@host# set nested-application my-nested-app signature
my-nested-app-sig:Social-Website member m01
```

6. Set the context to be used for matching the application.

```
[edit services application identification]
user@host# set nested-application my-nested-app signature
my-nested-app-sig:Social-Website member m01 context http-header-host
```

7. Set the pattern to match.

```
[edit services application identification]
user@host# set nested-application my-nested-app signature
my-nested-app-sig:Social-Website member m01 pattern
"*(facebook\.com|fbcdn\.net)";
```

8. Set the direction in which to match traffic.

```
[edit services application identification]
user@host# set nested-application my-nested-app signature
my-nested-app-sig:Social-Website direction client-to-server
```

9. Set the maximum number of transactions for a match to occur to 3.

```
[edit services application identification]
user@host# set nested-application my-nested-app signature
my-nested-app-sig:Social-Website maximum-transactions 3
```

10. Set the matching order for this signature to 5.

```
[edit services application identification]
user@host# set nested-application my-nested-app signature
my-nested-app-sig:Social-Website order 5
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Junos OS Application Identification Custom Application Definitions on page 777
  - Understanding Junos OS Application Identification Application Package on page 770
  - Understanding Junos OS Application Identification Services on page 769

---

## Application System Cache

- Understanding the Application System Cache on page 783
- Deactivating Application System Cache Information for Application Identification (CLI Procedure) on page 783
- Understanding Application System Cache Information for Nested Application Identification on page 784

- Deactivating Application System Cache Information for Nested Application Identification (CLI Procedure) on page 784
- Verifying Application System Cache Statistics on page 784

## Understanding the Application System Cache

Application system cache (ASC) saves the mapping between an application type and the corresponding destination IP address, destination port, protocol type, and service.

Once an application is identified, its information is saved in the ASC so that only one pattern matching is required for an application running on a particular system, thereby expediting the identification process.

A mapping is saved in the ASC only if the matched signature contains both client-to-server and server-to-client patterns. This process protects the system from hackers who might send malicious packets through a legitimate server port so that it is interpreted as a different application.

By default, the ASC saves the mapping information for 3600 seconds. However, you can configure the cache timeout value by using the CLI.

To minimize the impact on performance, application system cache is refreshed only when Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) traffic triggers a cache lookup. Without a cache lookup, the entries in the ASC remain unchanged even after cache timeout.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Junos OS Application Identification Services on page 769
  - Verifying Application System Cache Statistics on page 556

## Deactivating Application System Cache Information for Application Identification (CLI Procedure)

Application caching is turned on by default. You can manually turn this caching off using the CLI.

```
user@host# set services application-identification no-application-system-cache
```

When you use the show command in the CLI operation mode for the application system cache, application information is displayed as follows:

```
user@host> show services application-identification application-system-cache
```

```
Vsys-ID IP address Port Protocol Service Application
0 5.0.0.1 80 TCP HTTP FACEBOOK
0 5.0.0.2 80 TCP HTTP NONE
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Verifying Application System Cache Statistics on page 556
  - Understanding Junos OS Application Identification Services on page 769

## Understanding Application System Cache Information for Nested Application Identification

Nested application identification information is saved in the application system cache (ASC) to improve performance. The ASC is updated when a different application is identified. The only circumstances in which nested application information is not cached are the following:

- The application system cache is turned off for nested application identification.
- The matched application signatures have only client-to-server members.
- There is no valid server-to-client response seen for a transaction. This is done to prevent an attacker from sending invalid client-to-server requests to poison the ASC.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Junos OS Application Identification Services on page 769
  - Verifying Application System Cache Statistics on page 556

## Deactivating Application System Cache Information for Nested Application Identification (CLI Procedure)

Caching for nested applications is turned on by default. You can manually turn this caching off using the CLI.

```
user@host# set services application-identification nested-application-settings
no-application-system-cache
```

When you use the show command in the CLI operation mode for the application system cache (ASC), nested application information is displayed as follows:

```
user@host>show services application-identification application-system-cache

Vsys-ID IP address Port Protocol Service Application
0 5.0.0.1 80 TCP HTTP FACEBOOK
0 5.0.0.2 80 TCP HTTP NONE
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Activating Junos OS Application Identification for Nested Applications (CLI Procedure) on page 776
  - Verifying Application System Cache Statistics on page 556
  - Understanding Junos OS Application Identification Services on page 769

## Verifying Application System Cache Statistics

**Purpose** Verify the application system cache (ASC) statistics.



NOTE: The application system cache will display the cache for application identification applications and nested applications.



**Action** From CLI operation mode, enter the **show services application-identification application-system-cache** command.

**Sample Output**

```

user@host> show services application-identification application-system-cache
Application System Cache Statistics:
Vsys-ID IP Address Port Protocol Service Application
0 5.0.0.1 65532 TCP FTP Unknown
Application System Cache statistics:

 Vsys-ID IP address Port Protocol Service Application
0 5.0.0.1 65532 tcp FTP Unknown
0 20.0.0.4 23 tcp TELNET Unknown
0 20.0.0.6 23 tcp TELNET Unknown
0 20.0.0.2 23 tcp TELNET Unknown
0 20.0.0.2 25 tcp SMTP Unknown
0 20.0.0.6 25 tcp SMTP Unknown
0 20.0.0.4 25 tcp SMTP Unknown
0 20.0.0.3 135 tcp MSRPC Unknown
0 20.0.0.5 139 tcp SMB Unknown
0 20.0.0.7 139 tcp SMB Unknown
0 20.0.0.3 143 tcp IMAP Unknown
0 20.0.0.5 143 tcp IMAP Unknown
0 20.0.0.3 139 tcp SMB Unknown
0 20.0.0.7 143 tcp IMAP Unknown
0 20.0.0.3 80 tcp HTTP Unknown
0 20.0.0.5 80 tcp HTTP FACEBOOK
0 20.0.0.7 80 tcp HTTP ORKUT

```

**Meaning** The output shows a summary of the ASC statistics information. Verify the following information:

- Vsys-ID—Displays the virtual system identification number.
- IP address—Displays the destination address.
- Port—Displays the destination port on the server.
- Service—Displays the name of the service or application identified on the destination port.

For a complete description of **show security idp application-identification application-system-cache** output, see the *Junos OS CLI Reference*.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding the Application System Cache on page 783
  - Disabling Junos OS Application Identification (CLI Procedure) on page 775

## Memory and Session Limits

- Understanding Memory and Session Limit Settings for Junos OS Application Identification Services on page 786
- Example: Setting Memory and Session Limits for Junos OS Application Identification Services (CLI) on page 787

### Understanding Memory and Session Limit Settings for Junos OS Application Identification Services

You can configure settings to limit the number of sessions running application identification and also limit memory usage for application identification.

- **Memory limit for a session**—You can configure the maximum amount of memory bytes that can be used to save packets for application identification for one TCP or UDP session. You can also configure a limit for global memory usage for application identification. Application identification is disabled for a session after the system reaches the specified memory limit for the session. However, application identification continues to match patterns. The matched application is saved to cache so that the next session can use it. This protects the system from attackers trying to bypass application identification by purposely sending large client-to-server packets.
- **Number of sessions**—You can configure the maximum number of sessions that can run application identification at the same time. Application identification is disabled after the system reaches the specified number of sessions. You limit the number of sessions so that you can prevent a denial-of-service (DoS) attack, which occurs when too many connection requests overwhelm and exhaust all the allocated resources on the system.

Table 75 on page 786 shows the session capacity for a central point (CP) for SRX3400, SRX3600, SRX5600, and SRX5800 devices.

**Table 75: Maximum CP Sessions for Application Identification Services**

| SRX Series Devices | Maximum Sessions | Central Point (CP) |
|--------------------|------------------|--------------------|
| SRX3400            | 2.25 million     | Combo-mode CP      |
| SRX3600            | 2.25 million     | Combo-mode CP      |
| SRX5600            | 10 million       | Full CP            |
|                    | 2.25 million     | Combo-mode CP      |
| SRX5800            | 10 million       | Full CP            |
|                    | 2.25 million     | Combo-mode CP      |

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Junos OS Application Identification Services on page 769

- Example: Setting Memory and Session Limits for IDP Application Identification (CLI) on page 558

## Example: Setting Memory and Session Limits for Junos OS Application Identification Services (CLI)

The configuration instructions in this topic describe how to configure memory and session limits for application identification.

Before you begin, make sure that you have completed the following:

1. Configure network interfaces. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Download the application package. See “Updating Junos OS Application Identification Extracted Application Package Overview” on page 771.

In the configuration instructions for this example, you configure the limit so that only 600 sessions can run application identification at the same time. You also configure 5000 bytes as the maximum amount of memory that can be used for saving packets for application identification for one TCP session.

To configure memory and session limits for application identification:

1. Specify the session limit for application identification. In the following statement you set the maximum number of sessions that can run application identification at the same time as **600**:  
  
`user@host# set services application-identification max-sessions 600`
2. Specify the memory limit for application identification. In the following statement you configure a maximum of **5000** bytes to save packets for application identification:  
  
`user@host# set services application-identification max-tcp-session-packet-memory 5000`
3. If you are finished configuring the device, commit the configuration.
4. From configuration mode in the CLI, enter the **show services application-identification** command to verify the configuration. For more information, see the *Junos OS CLI Reference*.

### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Memory and Session Limit Settings for IDP Application Identification on page 557
- Understanding Junos OS Application Identification Services on page 769



# AppTrack Application Tracking

- Understanding AppTrack on page 789
- AppTrack Usage on page 790

## Understanding AppTrack

---

AppTrack, an application tracking tool, provides statistics for analyzing bandwidth usage of your network. When enabled, AppTrack collects byte, packet, and duration statistics for application flows in the specified zone. By default, when each session closes, AppTrack generates a message that provides the byte and packet counts and duration of the session, and sends it to the host device. The Security Threat Response Manager (STRM) retrieves the data and provides flow-based application visibility.

AppTrack messages are similar to session logs and use syslog or structured syslog formats. The message also includes an application field for the session. If AppTrack identifies a custom-defined application and returns an appropriate name, the custom application name is included in the log message. (If the application identification process fails or has not yet completed when an update message is triggered, the message specifies **none** in the application field.)

If you enable AppTrack for a zone and specify a **session-update-interval** time, whenever a packet is received, AppTrack checks whether the time since the start of the session or since the last update is greater than the update interval. If so, AppTrack updates the counts and sends an update message to the host. If a short-lived session starts and ends within the update interval, AppTrack generates a message only at session close.

When you want the initial update message to be sent earlier than the specified update interval, use the **first-update-interval**. The **first-update-interval** lets you enter a shorter interval for the first update only. Alternatively, you can generate the initial update message at session start by using the **first-update** option.



**NOTE:** If you specify both the **first-update** option and the **first-update-interval** option, AppTrack sends an update message when the session begins. In this case, the **first-update-interval** value is ignored, and a second message is sent when the next full update interval has elapsed.

---

The close message updates the statistics for the last time and provides an explanation for the session closure. The following codes are used:

**TCP RST**—RST received from either end.

**TCP FIN**—FIN received from either end.

**Response received**—Response received for a packet request (such as **icmp req-reply**).

**ICMP error**—ICMP error received (such as **dest unreachable**).

**Aged out**—Session aged out.

**ALG**—ALG closed the session.

**IDP**—IDP closed the session.

**Parent closed**—Parent session closed.

**CLI**—Session cleared by a CLI statement.

**Policy delete**—Policy marked for deletion.

---

## AppTrack Usage

- Example: Configuring AppTrack (CLI) on page 790
- Example: Verifying AppTrack Operation (CLI) on page 791

### Example: Configuring AppTrack (CLI)

The following example enables AppTrack. The option selections specify that the first message is generated 1 minute after session start and that update messages are sent every 5 minutes after that until the session ends. A final message is sent at session end.

1. Configure the remote syslog device to receive AppTrack messages.

```
[edit]
user@host# set security log format syslog
user@host# set security log source-address 5.0.0.254
user@host# set security log stream idpdata host 5.0.0.1
```

2. Navigate to the security level of the hierarchy.

```
[edit]
user@host# edit security
```

3. Enable AppTrack for the security zone trust.

```
[edit security]
user@host# set zone security-zone trust application-tracking
```

4. Generate update messages every 5 minutes.

```
[edit security]
user@host# set application-tracking session-update-interval 5
```

5. Generate the first message 1 minute after session start.

```
[edit security]
user@host# set application-tracking first-update-interval 1
```

Alternatively, to generate a message at session start and send update messages every 5 minutes after that, you could use the **first-update** option instead of the **first-update-interval** option.

```
[edit security]
user@host# set application-tracking first-update
```



**NOTE:** If you specify both the **first-update** option and the **first-update-interval** option, the **first-update-interval** value is ignored.

6. To verify that AppTrack is enabled and to check your settings, navigate to the top of the hierarchy and display the AppTrack configuration. In the following example, the application-tracking portion of the configuration listing shows 5-minute and 1-minute settings for the update intervals as configured in the previous steps.

```
user@host# top
[edit]
user@host# show security application-tracking

...
security {
 ...
 application-tracking {
 session-update-interval 5; #5 minutes
 first-update-interval 1; #1 minute
 }
 ...
}
```

7. If you are finished configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

For command option descriptions and values, see the *Junos OS CLI Reference*.

For general information about managing system log files, see the *Junos OS Administration Guide for Security Devices*.

### Example: Verifying AppTrack Operation (CLI)

The following examples provide two ways to monitor AppTrack operation.

- View AppTrack counters periodically to monitor tracking.

```
user@host> show security application-tracking counters
```

| AVT counters:           | Value |
|-------------------------|-------|
| Session create messages | 1     |
| Session close messages  | 1     |
| Session volume updates  | 0     |
| Failed messages         | 0     |

- Compare byte and packet counts in logged messages with the session statistics from the **show** command output.

```
user@host> show security flow session
```

```
Flow Sessions on FPC6 PIC0:
```

```
Session ID: 120000044, Policy name: policy-in-out/4, Timeout: 1796, Valid
In: 4.0.0.1/39075 --> 5.0.0.1/21;tcp, If: ge-0/0/0.0, Pkts: 22, Bytes:
1032
Out: 5.0.0.1/21 --> 4.0.0.1/39075;tcp, If: ge-0/0/1.0, Pkts: 24, Bytes:
1442
```

```
Valid sessions: 1
Pending sessions: 0
Invalidated sessions: 0
Sessions in other states: 0
Total sessions: 1
```

Byte and packet totals in the session statistics should approximate the counts logged by AppTrack but might not be exactly the same. AppTrack counts only incoming bytes and packets. System-generated packets are not included in the total, and dropped packets are not deducted.

For command option descriptions and values, see the *Junos OS CLI Reference*.

For general information about monitoring events and managing system log files, see the *Junos OS Administration Guide for Security Devices*.



## PART 11

# Chassis Cluster

- Chassis Cluster on page 795



## CHAPTER 41

# Chassis Cluster

- Chassis Cluster Overview on page 795
- Understanding Chassis Cluster Formation on page 796
- Chassis Cluster Redundancy Groups on page 797
- Chassis Cluster Redundant Ethernet Interfaces on page 817
- Conditional Route Advertising in a Chassis Cluster on page 825
- Chassis Cluster Control Plane on page 828
- Chassis Cluster Data Plane on page 838
- Consequences of Enabling Chassis Cluster on page 845
- Building a Chassis Cluster on page 857
- Chassis Cluster Upgrades on page 873
- Disabling Chassis Cluster on page 877
- Understanding Multicast Routing on a Chassis Cluster on page 877
- Asymmetric Chassis Cluster Deployment on page 878
- Active/Passive Chassis Cluster Deployment (J Series Devices) on page 883
- Active/Passive Chassis Cluster Deployment (SRX Series Devices) on page 888
- Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 915
- Limitations of Chassis Clustering on page 923

## Chassis Cluster Overview

---

Chassis clustering provides network node redundancy by grouping a pair of the same kind of supported SRX Series devices or J Series devices into a cluster. The devices must be running Junos OS. The control ports on the respective nodes are connected to form a control plane that synchronizes configuration and kernel state to facilitate the high availability of interfaces and services. Similarly, the data plane on the respective nodes is connected over the fabric ports to form a unified data plane. The fabric link allows for the management of cross-node flow processing and for the management of session redundancy.

The control plane software operates in active or backup mode. When configured as a chassis cluster, the two nodes back up each other, with one node acting as the primary device and the other as the secondary device, ensuring stateful failover of processes and

services in the event of system or hardware failure. If the primary device fails, the secondary device takes over processing of traffic.

The data plane software operates in active/active mode. In a chassis cluster, session information is updated as traffic traverses either device and this information is transmitted between the nodes over the fabric link to guarantee that established sessions are not dropped when a failover occurs. In active/active mode, it is possible for traffic to ingress the cluster on one node and egress from the other node.



**NOTE:** Devices running IP version 6 (IPv6) can be deployed only in active/backup chassis cluster configurations.

---

Chassis cluster functionality includes:

- Resilient system architecture, with a single active control plane for the entire cluster and multiple Packet Forwarding Engines. This architecture presents a single device view of the cluster.
- Synchronization of configuration and dynamic runtime states between nodes within a cluster.
- Monitoring of physical interfaces, and failover if the failure parameters cross a configured threshold.

The different states that a cluster can be in at any given instant are as follows: hold, primary, secondary-hold, secondary, ineligible, and disabled. A state transition can be triggered because of any event, such as interface monitoring, SPU monitoring, failures, and manual failovers.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Chassis Cluster Formation on page 796
  - Understanding Chassis Cluster Redundancy Groups on page 797
  - Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817

---

## Understanding Chassis Cluster Formation

---

To form a chassis cluster, a pair of the same kind of supported SRX Series devices or J Series devices are combined to act as a single system that enforces the same overall security. For SRX5600 and SRX5800 chassis clusters, the placement and type of Services Processing Cards (SPCs) must match in the two clusters. For SRX3400 and SRX3600 chassis clusters, the placement and type of SPCs, I/O cards (IOCs), and Network Processing Cards (NPCs) must match in the two devices.

For J Series chassis clusters, although the devices must be the same kind, they can contain different Physical Interface Modules (PIMs).

When a device joins a cluster, it becomes a node of that cluster. With the exception of unique node settings and management IP addresses, nodes in a cluster share the same configuration.

You can deploy up to 15 chassis clusters in a Layer 2 domain. Clusters and nodes are identified in the following way:

- A cluster is identified by a *cluster ID* (**cluster-id**) specified as a number from 1 through 15.
- A cluster node is identified by a *node ID* (**node**) specified as a number from 0 to 1.

**Related Topics**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Chassis Cluster Overview on page 795
- Understanding Chassis Cluster Redundancy Groups on page 797
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Understanding the Chassis Cluster Control Plane on page 828
- Understanding the Chassis Cluster Data Plane on page 838
- Understanding What Happens When Chassis Cluster Is Enabled on page 845

## Chassis Cluster Redundancy Groups

---

- Understanding Chassis Cluster Redundancy Groups on page 797
- Chassis Cluster Redundancy Groups 0 Through 128 on page 798
- Chassis Cluster Redundancy Group Interface Monitoring on page 804
- Chassis Cluster Redundancy Group IP Address Monitoring on page 806
- Understanding Chassis Cluster Monitoring of Global-Level Objects on page 810
- Chassis Cluster Redundancy Group Failover on page 812

## Understanding Chassis Cluster Redundancy Groups

Chassis clustering provides high availability of interfaces and services through redundancy groups and primacy within groups.

A redundancy group is an abstract construct that includes and manages a collection of objects. A redundancy group contains objects on both nodes. A redundancy group is primary on one node and backup on the other at any time. When a redundancy group is said to be primary on a node, its objects on that node are active.

Redundancy groups are independent units of failover. Each redundancy group fails over from one node to the other independent of other redundancy groups. When a redundancy group fails over, all its objects fail over together.

Three things determine the primacy of a redundancy group: the priority configured for the node, the node ID (in case of tied priorities), and the order in which the node comes

up. If a lower priority node comes up first, then it will assume the primacy for a redundancy group (and will stay as primary if preempt is not enabled).

A chassis cluster can include many redundancy groups, some of which might be primary on one node and some of which might be primary on the other. Alternatively, all redundancy groups can be primary on a single node. One redundancy group's primacy does not affect another redundancy group's primacy. You can create up to 128 redundancy groups.



**NOTE:** The maximum number of redundancy groups is equal to the number of redundant Ethernet interfaces that you configure.

---

You can configure redundancy groups to suit your deployment. You configure a redundancy group to be primary on one node and backup on the other node. You specify the node on which the group is primary by setting priorities for both nodes within a redundancy group configuration. The node with the higher priority takes precedence, and the redundancy group's objects on it are active.

If a redundancy group is configured so that both nodes have the same priority, the node with the lowest node ID number always takes precedence, and the redundancy group is primary on it. In a two-node cluster, node 0 always takes precedence in a priority tie.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 798
- Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 799
- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 804
- Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 806
- Understanding Chassis Cluster Redundancy Group Failover on page 812
- Understanding Chassis Cluster Formation on page 796

## Chassis Cluster Redundancy Groups 0 Through 128

- Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 798
- Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 799
- Example: Configuring Chassis Cluster Redundancy Groups (CLI) on page 803
- Verifying Chassis Cluster Redundancy Group Status on page 804

### Understanding Chassis Cluster Redundancy Group 0: Routing Engines

When you initialize a device in chassis cluster mode, the system creates a redundancy group referred to as redundancy group 0. Redundancy group 0 manages the primacy and failover between the Routing Engines on each node of the cluster. As is the case for all redundancy groups, redundancy group 0 can be primary on only one node at a time. The node on which redundancy group 0 is primary determines which Routing Engine is active in the cluster. A node is considered the primary node of the cluster if its Routing Engine is the active one.

The redundancy group 0 configuration specifies the priority for each node. The following priority scheme determines redundancy group 0 primacy. Note that the three-second value is the interval if the default **heartbeat-threshold** and **heartbeat-interval** values are used.

- The node that comes up first (at least three seconds prior to the other node) is the primary node.
- If both nodes come up at the same time (or within three seconds of each other):
  - The node with the higher configured priority is the primary node.
  - If there is a tie (either because the same value was configured or because default settings were used), the node with the lower node ID (node 0) is the primary node.

The previous priority scheme applies to redundancy groups *x* (redundancy groups numbered 1 through 128) as well, provided preempt is not configured.

You cannot enable preemption for redundancy group 0. If you want to change the primary node for redundancy group 0, you must do a manual failover.



**CAUTION:** Be cautious and judicious in your use of redundancy group 0 manual failovers. A redundancy group 0 failover implies a Routing Engine failover, in which case all processes running on the primary node are killed and then spawned on the new master Routing Engine. This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 799
- Example: Configuring Chassis Cluster Redundancy Groups (CLI) on page 803
- Verifying Chassis Cluster Redundancy Group Status on page 804
- Understanding Chassis Cluster Redundancy Group Failover on page 812
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Understanding Chassis Cluster Formation on page 796

#### Understanding Chassis Cluster Redundancy Groups 1 Through 128

You can configure one or more redundancy groups numbered 1 through 128, referred to as redundancy group *x*. The maximum number of redundancy groups is equal to the number of redundant Ethernet interfaces that you configure (see Table 78 on page 818). Each redundancy group *x* acts as an independent unit of failover and is primary on only one node at a time.

Each redundancy group *x* contains one or more redundant Ethernet interfaces. A redundant Ethernet interface is a pseudointerface that contains at minimum a pair of physical Gigabit Ethernet interfaces or a pair of Fast Ethernet interfaces. If a redundancy group is active on node 0, then the child links of all the associated redundant Ethernet interfaces

on node 0 are active. If the redundancy group fails over to node 1, then the child links of all redundant Ethernet interfaces on node 1 become active.

On SRX Series and J Series chassis clusters, you can configure multiple redundancy groups to load-share traffic across the cluster. For example, you can configure some redundancy groups *x* to be primary on one node and some redundancy groups *x* to be primary on the other node. You can also configure a redundancy group *x* in a one-to-one relationship with a single redundant Ethernet interface to control which interface traffic flows through.

The traffic for a redundancy group is processed on the node where the redundancy group is active. Because more than one redundancy group can be configured, it is possible that the traffic from some redundancy groups will be processed on one node while the traffic for other redundancy groups is processed on the other node (depending on where the redundancy group is active). Multiple redundancy groups make it possible for traffic to arrive over an ingress interface of one redundancy group and over an egress interface that belongs to another redundancy group. In this situation, the ingress and egress interfaces might not be active on the same node. When this happens, the traffic is forwarded over the fabric link to the appropriate node.

When you configure a redundancy group *x*, you must specify a priority for each node to determine the node on which the redundancy group *x* is primary. The node with the higher priority is selected as primary. The primacy of a redundancy group *x* can fail over from one node to the other. When a redundancy group *x* fails over to the other node, its redundant Ethernet interfaces on that node are active and their interfaces are passing traffic.

Table 76 on page 800 gives an example of redundancy group *x* in an SRX Series chassis cluster and indicates the node on which the group is primary. It shows the redundant Ethernet interfaces and their interfaces configured for redundancy group *x*.



**NOTE:** SRX210 devices have both Gigabit Ethernet ports and Fast Ethernet ports.

**Table 76: Example of Redundancy Groups for an SRX Series Chassis Cluster (SRX3000 and SRX5000 Lines)**

| Group              | Primary | Priority    | Objects                  | Interface (Node 0) | Interface (Node 1) |
|--------------------|---------|-------------|--------------------------|--------------------|--------------------|
| Redundancy group 0 | Node 0  | Node 0: 254 | Routing Engine on node 0 | —                  | —                  |
|                    |         | Node 1: 2   | Routing Engine on node 1 | —                  | —                  |



**Table 76: Example of Redundancy Groups for an SRX Series Chassis Cluster (SRX3000 and SRX5000 Lines) (*continued*)**

| Group              | Primary | Priority    | Objects                        | Interface (Node 0) | Interface (Node 1) |
|--------------------|---------|-------------|--------------------------------|--------------------|--------------------|
| Redundancy group 1 | Node 0  | Node 0: 254 | Redundant Ethernet interface 0 | <b>ge-1/0/0</b>    | <b>ge-23/0/0</b>   |
|                    |         | Node 1: 2   | Redundant Ethernet interface 1 | <b>ge-1/3/0</b>    | <b>ge-23/3/0</b>   |
| Redundancy group 2 | Node 1  | Node 0: 2   | Redundant Ethernet interface 2 | <b>ge-2/0/0</b>    | <b>ge-24/0/0</b>   |
|                    |         | Node 1: 254 | Redundant Ethernet interface 3 | <b>ge-2/3/0</b>    | <b>ge-24/3/0</b>   |
| Redundancy group 3 | Node 0  | Node 0: 254 | Redundant Ethernet interface 4 | <b>ge-3/0/0</b>    | <b>ge-25/0/0</b>   |
|                    |         | Node 1: 2   | Redundant Ethernet interface 5 | <b>ge-3/3/0</b>    | <b>ge-25/3/0</b>   |

As the example for an SRX Series chassis cluster in Table 76 on page 800 shows:

- The Routing Engine on node 0 is active because redundancy group 0 is primary on node 0. (The Routing Engine on node 1 is passive, serving as backup.)
- Redundancy group 1 is primary on node 0. Interfaces **ge-1/0/0** and **ge-1/3/0** belonging to redundant Ethernet interface 0 and redundant Ethernet interface 1 are active and handling traffic.
- Redundancy group 2 is primary on node 1. Interfaces **ge-24/0/0** and **ge-24/3/0** belonging to redundant Ethernet interface 2 and redundant Ethernet interface 3 are active and handling traffic.
- Redundancy group 3 is primary on node 0. Interfaces **ge-3/0/0** and **ge-3/3/0** belonging to redundant Ethernet interface 4 and redundant Ethernet interface 5 are active and handling traffic.

Table 77 on page 802 gives an example of redundancy groups *x* in a J Series chassis cluster and indicates the node on which each group is primary. It shows the redundant Ethernet interfaces and their interfaces configured for each redundancy group *x*.

Table 77: Example of Redundancy Groups for a J Series Chassis Cluster

| Group              | Primary | Priority    | Objects                        | Interface (Node 0) | Interface (Node 1) |
|--------------------|---------|-------------|--------------------------------|--------------------|--------------------|
| Redundancy group 0 | Node 1  | Node 0: 50  | Routing Engine on node 0       | —                  | —                  |
|                    |         | Node 1: 100 | Routing Engine on node 1       | —                  | —                  |
| Redundancy group 1 | Node 1  | Node 0: 50  | Redundant Ethernet interface 0 | <b>fe-1/0/0</b>    | <b>fe-8/0/0</b>    |
|                    |         | Node 1: 100 | Redundant Ethernet interface 1 | <b>fe-1/0/1</b>    | <b>fe-8/0/1</b>    |
| Redundancy group 2 | Node 1  | Node 0: 50  | Redundant Ethernet interface 2 | <b>ge-2/0/0</b>    | <b>ge-9/0/0</b>    |
|                    |         | Node 1: 100 | Redundant Ethernet interface 3 | <b>ge-2/0/1</b>    | <b>ge-9/0/1</b>    |
| Redundancy group 3 | Node 0  | Node 0: 100 | Redundant Ethernet interface 4 | <b>ge-3/0/0</b>    | <b>ge-10/0/0</b>   |
|                    |         | Node 1: 50  | Redundant Ethernet interface 5 | <b>ge-3/0/1</b>    | <b>ge-10/0/1</b>   |

As the example for a J Series chassis cluster in Table 77 on page 802 shows:

- The Routing Engine on node 1 is active because redundancy group 0 is primary on node 1. (The Routing Engine on node 0 is passive, serving as backup.)
- Redundancy group 1 is primary on node 1. Interfaces **fe-8/0/0** and **fe-8/0/1** belonging to redundant Ethernet interface 0 and redundant Ethernet interface 1 are active and handling traffic.
- Redundancy group 2 is primary on node 1. Interfaces **ge-9/0/0** and **ge-9/0/1** belonging to redundant Ethernet interface 2 and redundant Ethernet interface 3 are active and handling traffic.
- Redundancy group 3 is primary on node 0. Interfaces **ge-3/0/0** and **ge-3/0/1** belonging to redundant Ethernet interface 4 and redundant Ethernet interface 5 are active and handling traffic.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 798

- Example: Configuring Chassis Cluster Redundancy Groups (CLI) on page 803
- Verifying Chassis Cluster Redundancy Group Status on page 804
- Understanding Chassis Cluster Redundancy Group Failover on page 812
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Understanding Chassis Cluster Formation on page 796

### Example: Configuring Chassis Cluster Redundancy Groups (CLI)

A redundancy group is an abstract entity that includes and manages a collection of objects. A redundancy group can be primary on only one node at a time.

Before you begin, complete the following tasks:

- Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
- Example: Configuring the Chassis Cluster Fabric (CLI) on page 842

Before you can create redundant Ethernet interfaces, you must first create their redundancy groups.

Use the following command in configuration mode to specify the number of gratuitous ARP requests that an interface can send to notify other network devices of its presence after the redundancy group it belongs to has failed over:

```
{primary:node1}
user@host# set chassis cluster redundancy-group 1 gratuitous-arp-count 4
```

Use the following command in configuration mode to identify an interface to be monitored by a specific redundancy group and give it a weight. You can configure a redundancy group to monitor any interfaces, not only those belonging to its redundant Ethernet interfaces.

```
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-3/1/1/1 weight 100
```

Use the following commands in configuration mode to specify a redundancy group's priority for primacy on each node of the cluster. The higher number takes precedence.

```
{primary:node1}
user@host# set chassis cluster redundancy-group 1 node 1 priority 100
{secondary:node0}
user@host# set chassis cluster redundancy-group 1 node 0 priority 200
```

Use the following command in configuration mode to specify if a node with a higher priority can initiate a failover to become primary for the redundancy group:

```
{primary:node1}
user@host# set chassis cluster redundancy-group 1 preempt
```

Use the following command in configuration mode to specify the minimum interval to be allowed between back-to-back failovers for the redundancy group:

```
{primary:node1}
user@host# set chassis cluster redundancy-group 1 hold-down-interval 6
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 798
  - Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 799
  - Verifying Chassis Cluster Redundancy Group Status on page 804
  - Understanding Chassis Cluster Redundancy Group Failover on page 812
  - Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
  - Understanding Chassis Cluster Formation on page 796

### Verifying Chassis Cluster Redundancy Group Status

**Purpose** Display the failover status of a chassis cluster redundancy group.

**Action** From the CLI, enter the **show chassis cluster status redundancy-group** command:

```
{primary:node1}
user@host> show chassis cluster status redundancy-group 2
Cluster ID: 14
 Node name Priority Status Preempt Manual failover

Redundancy-Group: 2, Failover count: 1
 node0 50 secondary no no
 node1 100 primary no no
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Configuring Chassis Cluster Redundancy Groups (CLI) on page 803
  - Verifying Chassis Cluster Interfaces on page 821
  - Verifying a Chassis Cluster Configuration on page 869

## Chassis Cluster Redundancy Group Interface Monitoring

- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 804
- Example: Configuring Chassis Cluster Interface Monitoring (CLI) on page 805

### Understanding Chassis Cluster Redundancy Group Interface Monitoring

For a redundancy group to automatically fail over to another node, its interfaces must be monitored. When you configure a redundancy group, you can specify a set of interfaces that the redundancy group is to monitor for status (or “health”) to determine whether the interface is up or down. A monitored interface can be a child interface of any of its redundant Ethernet interfaces. When you configure an interface for a redundancy group to monitor, you give it a weight.

Every redundancy group has a threshold tolerance value initially set to **255**. When an interface monitored by a redundancy group becomes unavailable, its weight is subtracted from the redundancy group's threshold. When a redundancy group's threshold reaches 0, it fails over to the other node. For example, if redundancy group 1 was primary on node 0, on the threshold-crossing event, redundancy group 1 becomes primary on node 1. In this case, all the child interfaces of redundancy group 1's redundant Ethernet interfaces begin handling traffic.

A redundancy group failover occurs because the cumulative weight of the redundancy group's monitored interfaces has brought its threshold value to 0. When the monitored interfaces of a redundancy group on both nodes reach their thresholds at the same time, the redundancy group is primary on the node with the lower node ID, in this case node 0.



**NOTE:** If you want to dampen the failovers occurring because of interface monitoring failures, use the `hold-down-interval` statement.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 798
- Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 799
- Example: Configuring Chassis Cluster Interface Monitoring (CLI) on page 805
- Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 806
- Understanding Chassis Cluster Redundancy Group Failover on page 812
- Understanding Chassis Cluster Monitoring of Global-Level Objects on page 810

#### Example: Configuring Chassis Cluster Interface Monitoring (CLI)

Redundancy group failover is triggered by the results from monitoring the health of interfaces that belong to the redundancy group. When you assign a weight to an interface to be monitored, the system monitors the interface for availability. If a physical interface fails, the weight is deducted from the corresponding redundancy group's threshold. Every redundancy group has a threshold of 255. If the threshold hits 0, a failover is triggered. Failover is triggered even if the redundancy group is in manual failover mode and preempt is not enabled.

Use the following command to set interface monitoring on `ge-7/0/3`:

```
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-7/0/3 weight
255
```

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 798
- Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 799
- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 804
- Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 806

- Understanding Chassis Cluster Redundancy Group Failover on page 812
- Understanding Chassis Cluster Monitoring of Global-Level Objects on page 810

## Chassis Cluster Redundancy Group IP Address Monitoring

- Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 806
- Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring (CLI) on page 808

### Understanding Chassis Cluster Redundancy Group IP Address Monitoring

Redundancy group IP address monitoring checks end-to-end connectivity and allows a redundancy group to fail over because of the inability of a redundant Ethernet interface (known as a **reth**) to reach a configured IP address. Redundancy groups on both devices in a cluster can be configured to monitor specific IP addresses to determine whether an upstream device in the network is reachable. The redundancy group can be configured such that if the monitored IP address becomes unreachable, the redundancy group will fail over to its backup to maintain service. The primary difference between this monitoring feature and interface monitoring is that IP address monitoring allows for failover when the interface is still up but the network device it is connected to is not reachable for some reason. It may be possible under those circumstances for the other node in the cluster to route traffic around the problem.



**NOTE:** If you want to dampen the failovers occurring because of IP address monitoring failures, use the **hold-down-interval** statement.

---

IP address monitoring configuration allows you to set not only the address to monitor and its failover weight but also a global IP address monitoring threshold and weight. Only after the IP address monitoring global-threshold is reached because of cumulative monitored address reachability failure will the IP address monitoring global-weight value be deducted from the redundant group's failover threshold. Thus, multiple addresses can be monitored simultaneously as well as monitored to reflect their importance to maintaining traffic flow. Also, the threshold value of an IP address that is unreachable and then becomes reachable again will be restored to the monitoring threshold. This will not, however, cause a fallback unless the preempt option has been enabled.

When configured, the IP address monitoring failover value (global-weight) is considered along with interface monitoring—if set—and built-in failover monitoring, including SPU monitoring, cold-sync monitoring, and NPC monitoring (on supported platforms). The main IP addresses that should be monitored are router gateway addresses to ensure that valid traffic coming into the services gateway can be forwarded to the appropriate network router.

One Services Processing Unit (SPU) or Packet Forwarding Engine (PFE) per node is designated to send Internet Control Message Protocol (ICMP) ping packets for the monitored IP addresses on the cluster. The primary PFE sends ping packets using Address Resolution Protocol (ARP) requests resolved by the Routing Engine (RE). The source for these pings is the redundant Ethernet interface MAC and IP addresses. The secondary PFE resolves ARP requests for the monitored IP address itself. The source for these pings

is the physical child MAC address and a secondary IP address configured on the redundant Ethernet interface. For the ping reply to be received on the secondary interface, the I/O card (IOC), central PFE processor, or Flex IOC adds both the physical child MAC address and the redundant Ethernet interface MAC address to its MAC table. The secondary PFE responds with the physical child MAC address to ARP requests sent to the secondary IP address configured on the redundant Ethernet interface.

The default interval to check the reachability of a monitored IP address is once per second. The interval can be adjusted using the **retry-interval** command. The default number of permitted consecutive failed ping attempts is **5**. The number of allowed consecutive failed ping attempts can be adjusted using the **retry-count** command. After failing to reach a monitored IP address for the configured number of consecutive attempts, the IP address is determined to be unreachable and its failover value is deducted from the redundancy group's global-threshold.

Once the IP address is determined to be unreachable, its weight is deducted from the global-threshold. If the recalculated global-threshold value is not 0, the IP address is marked unreachable, but the global-weight is not deducted from the redundancy group's threshold. If the redundancy group IP monitoring global-threshold reaches 0 and there are unreachable IP addresses, the redundancy group will continuously fail over and fail back between the nodes until either an unreachable IP address becomes reachable or a configuration change removes unreachable IP addresses from monitoring. Note that both default and configured **hold-down-interval** failover dampening is still in effect.

Every redundancy group *x* has a threshold tolerance value initially set to **255**. When an IP address monitored by redundancy group *x* becomes unavailable, its weight is subtracted from the redundancy group *x*'s threshold. When redundancy group *x*'s threshold reaches 0, it fails over to the other node. For example, if redundancy group 1 was primary on node 0, on the threshold-crossing event, redundancy group 1 becomes primary on node 1. In this case, all the child interfaces of redundancy group 1's redundant Ethernet interfaces begin handling traffic.

A redundancy group *x* failover occurs because the cumulative weight of the redundancy group *x*'s monitored IP addresses and other monitoring has brought its threshold value to 0. When the monitored IP addresses of redundancy group *x* on both nodes reach their thresholds at the same time, redundancy group *x* is primary on the node with the lower node ID, which is typically node 0.

Monitoring can be accomplished only if the IP address is reachable on a redundant Ethernet interface (known as a **reth** in CLI commands and interface listings), and IP addresses cannot be monitored over a tunnel. For an IP address to be monitored through a redundant Ethernet interface on a secondary cluster node, the interface must have a secondary IP address configured. IP address monitoring cannot be used on a chassis cluster running in transparent mode. The maximum number of monitoring IPs that can be configured per cluster is 64 for the SRX5000 line and 32 for the SRX3000 line.



**NOTE:** Redundancy group IP address monitoring is not supported for IPv6 destinations in this release.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 798
  - Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 799
  - Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 804
  - Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring (CLI) on page 808
  - Understanding Chassis Cluster Redundancy Group Failover on page 812
  - Understanding Chassis Cluster Monitoring of Global-Level Objects on page 810

### Example: Configuring Chassis Cluster Redundancy Group IP Address Monitoring (CLI)

Redundancy groups can be configured to monitor key upstream resources by pinging specific IP addresses reachable through redundant Ethernet interfaces on either node in a cluster. Global threshold, weight, retry-interval, and retry-count parameters can be set on a per-redundancy-group basis. IP address monitoring weights are configured against the redundancy group global-threshold, so when a monitored IP address becomes unreachable, the weight of that monitored IP address is deducted from the redundancy group IP address monitoring global-threshold. Only when the global-threshold reaches 0 will the global-weight be deducted from the redundancy group threshold. Retry-interval and retry-count settings determine the behavior of ping attempts for all IP addresses monitored by the redundancy group.

This section gives CLI configuration examples for:

- Configuring the IP address monitoring global-weight value
- Configuring the IP address monitoring global-threshold value
- Configuring the IP address retry-interval value
- Configuring the IP address retry-count value
- Configuring an IP address to monitor

Use the following command in configuration mode to set a global monitoring weight for all IP address monitoring by the redundancy group. The global-weight is deducted from the redundancy group's threshold when the IP address monitoring global-threshold value reaches 0. If you do not set a global-weight, the default global-weight is 255. Note that every redundancy group has a threshold of 255. If the threshold drops to 0 or below because of failures of monitored objects, the priority of that redundancy group on that node is reduced to 0. If the other node has a higher priority level at that point, it takes over and becomes the primary node.

```
{primary:node0}
user@host# set chassis cluster redundancy-group 1 ip-monitoring global-weight 100
```

Use the following command in configuration mode to set a global monitoring threshold for IP address monitoring by the redundancy group. The threshold applies cumulatively



to all IP addresses monitored by the redundancy group. If you do not set a `global-threshold`, the default `global-threshold` is `0`.

```
{primary:node0}
user@host# set chassis cluster redundancy-group 1 ip-monitoring global-threshold 200
```

Use the following command in configuration mode to set the ping interval for each IP address monitored by the redundancy group. The pings begin to be sent as soon as the configuration is committed. If you do not set a `retry-interval` value, the default interval is 1 second, which means that IP monitoring pings are sent to each monitored address every second.

```
{primary:node0}
user@host# set chassis cluster redundancy-group 1 ip-monitoring retry-interval 3
```

Use the following command in configuration mode to set the number of allowed consecutive ping failures for each IP address monitored by the redundancy group. If you do not set a `retry-count` value, the default count is `5`, which means that an IP address monitored by the redundancy group can fail to reply to a ping request five consecutive times before being considered unreachable.

```
{primary:node0}
user@host# set chassis cluster redundancy-group 1 ip-monitoring retry-count 10
```

Use the following command in configuration mode to specify an IP address to be monitored by a specific redundancy group through a specific redundant Ethernet interface (known as a **reth** in CLI commands and interface listings) and give it a weight. You can configure a redundancy group to monitor any IP address reachable through a redundant Ethernet interface, but you must name the interface, its logical unit number, and a secondary IP address for use as the IP monitoring ping source for the secondary node. Each of these elements is required to configure an IP address for monitoring. Note that the secondary IP address being assigned must be on the same subnet as the IP address of the redundant Ethernet interface logical unit and it should not be the same as the IP address on the redundant Ethernet interface.

```
{primary:node0}
user@host# set chassis cluster redundancy-group 1 ip-monitoring family inet 10.1.1.10
weight 100 interface reth1.0 secondary-ip-address 10.1.1.101
```

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 798
- Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 799
- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 804
- Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 806
- Understanding Chassis Cluster Redundancy Group Failover on page 812
- Understanding Chassis Cluster Monitoring of Global-Level Objects on page 810

## Understanding Chassis Cluster Monitoring of Global-Level Objects

There are various types of objects to monitor as you work with devices configured as chassis clusters, including global-level objects and objects that are specific to redundancy groups. This section describes the monitoring of global-level objects.

The SRX3000 and SRX5000 lines have one or more Services Processing Units (SPUs) that run on a Services Processing Card (SPC). All flow-based services run on the SPU. Other SRX Series devices and all J Series devices have a flow-based forwarding process, *flowd*, which forwards packets through the device.

- Understanding SPU Monitoring on page 810
- Understanding Flowd Monitoring on page 810
- Understanding Cold-Sync Monitoring on page 811

## Understanding SPU Monitoring

SPU monitoring tracks the health of the SPUs and of the central point (CP). The chassis manager on each SPC monitors the SPUs and the central point, and also maintains the heartbeat with the Routing Engine chassisd. In this hierarchical monitoring system, chassisd is the center for hardware failure detection. SPU monitoring is enabled by default.



**NOTE:** SPU monitoring is supported on the SRX3000 and SRX5000 lines.

---

Persistent SPU and central point failure on a node is deemed a catastrophic Packet Forwarding Engine (PFE) failure. In this case, the node's PFE is disabled in the cluster by reducing the priorities of redundancy groups *x* to 0.

- A central point failure triggers failover to the secondary node. The failed node's PFE, which includes all SPCs and all I/O cards (IOCs), is automatically restarted. If the secondary central point has failed as well, the cluster is unable to come up because there is no primary device. Only the data plane (redundancy group *x*) is failed over.
- A single, failed SPU causes failover of redundancy group *x* to the secondary node. All IOCs and SPCs on the failed node are restarted and redundancy group *x* is failed over to the secondary node. Failover to the secondary node is automatic without the need for user intervention. When the failed (former) primary node has its failing component restored, failback is determined by the preempt configuration for the redundancy group *x*. The interval for dead SPU detection is 30 seconds.

## Understanding Flowd Monitoring

Flowd monitoring tracks the health of the flowd process. Flowd monitoring is enabled by default.



**NOTE:** Flowd monitoring is supported on SRX210 and SRX100 devices. It is not supported on J Series devices.

---

Persistent flowd failure on a node is deemed a catastrophic Packet Forwarding Engine (PFE) failure. In this case, the node's PFE is disabled in the cluster by reducing the priorities of redundancy groups  $x$  to 0.

A failed flowd process causes failover of redundancy group  $x$  to the secondary node. Failover to the secondary node is automatic without the need for user intervention. When the failed (former) primary node has its failing component restored, failback is determined by the preempt configuration for the redundancy group  $x$ .

## Understanding Cold-Sync Monitoring

The process of synchronizing the data plane runtime objects (RTOs) on the startup of the SPUs or flowd is called *cold sync*. When all the RTOs are synchronized, the cold-sync process is complete, and the SPU or flowd on the node is ready to take over for the primary node, if needed. The process of monitoring the cold-sync state of all the SPUs or flowd on a node is called *cold-sync monitoring*. Keep in mind that when preempt is enabled, cold-sync monitoring prevents the node from taking over the mastership until the cold-sync process is completed for the SPUs or flowd on the node. Cold-sync monitoring is enabled by default.

When the node is rebooted, or when the SPUs or flowd come back up from failure, the priority for all the redundancy groups  $x$  is 0. When an SPU or flowd comes up, it tries to start the cold-sync process with its mirror SPU or flowd on the other node.

If this is the only node in the cluster, the priorities for all the redundancy groups  $x$  stay at 0 until a new node joins the cluster. Although the priority is at 0, the device can still receive and send traffic over its interfaces. A priority of 0 implies that it cannot fail over in case of a failure. When a new node joins the cluster, all the SPUs or flowd, as they come up, will start the cold-sync process with the mirror SPUs or flowd of the existing node.

When the SPU or flowd of a node that is already up detects the cold-sync request from the SPU or flowd of the peer node, it posts a message to the system indicating that the cold-sync process is complete. The SPUs or flowd of the newly joined node posts a similar message. However, they post this message only after all the RTOs are learned and cold-sync is complete. On receipt of completion messages from all the SPUs or flowd, the priority for redundancy groups  $x$  moves to the configured priority on each node if there are no other failures of monitored components, such as interfaces. This action ensures that the existing primary node for redundancy  $x$  groups always moves to the configured priority first. The node joining the cluster later moves to its configured priorities only after all its SPUs or flowd have completed their cold-sync process. This action in turn guarantees that the newly added node is ready with all the RTOs before it takes over mastership.

### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Group 0: Routing Engines on page 798
- Understanding Chassis Cluster Redundancy Groups 1 Through 128 on page 799
- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 804
- Understanding Chassis Cluster Redundancy Group IP Address Monitoring on page 806

- Understanding Chassis Cluster Redundancy Group Failover on page 812
- Understanding Chassis Cluster Formation on page 796

## Chassis Cluster Redundancy Group Failover

- Understanding Chassis Cluster Redundancy Group Failover on page 812
- Understanding Chassis Cluster Redundancy Group Manual Failover on page 813
- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 814
- Example: Configuring Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers (CLI) on page 816
- Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 816

### Understanding Chassis Cluster Redundancy Group Failover

Chassis cluster employs a number of highly efficient failover mechanisms that promote high availability to increase your system's overall reliability and productivity.

A redundancy group is a collection of objects that fail over as a group. Each redundancy group monitors a set of objects (physical interfaces), and each monitored object is assigned a weight. Each redundancy group has an initial threshold of **255**. When a monitored object fails, the weight of the object is subtracted from the threshold value of the redundancy group. When the threshold value reaches zero, the redundancy group fails over to the other node. As a result, all the objects associated with the redundancy group fail over as well. Graceful restart of the routing protocols enables the SRX Series device to minimize traffic disruption during a failover.

Because back-to-back redundancy group 0 failovers that occur too quickly can cause a cluster to exhibit unpredictable behavior, a dampening time between failovers is needed. On a failover, the previous primary node moves to the secondary-hold state and stays there until the hold-down interval expires, after which it moves to the secondary state.

The default dampening time is 300 seconds (5 minutes) for redundancy group 0 and is configurable to up to 1800 seconds with the **hold-down-interval** statement. Redundancy groups *x* (redundancy groups numbered 1 through 128) have a default dampening time of 1 second, with a range of 0 through 1800 seconds. The hold-down interval affects manual failovers, as well as automatic failovers associated with monitoring failures.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Group Interface Monitoring on page 804
- Understanding Chassis Cluster Redundancy Group Manual Failover on page 813
- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 814
- Example: Configuring Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers (CLI) on page 816
- Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 816

- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Understanding Chassis Cluster Monitoring of Global-Level Objects on page 810

### Understanding Chassis Cluster Redundancy Group Manual Failover

You can initiate a redundancy group *x* failover manually. A manual failover applies until a failback event occurs.

For example, suppose that you manually do a redundancy group 1 failover from node 0 to node 1. Then an interface that redundancy group 1 is monitoring fails, dropping the threshold value of the new primary redundancy group to zero. This event is considered a failback event, and the system returns control to the original redundancy group.

You can also initiate a redundancy group 0 failover manually if you want to change the primary node for redundancy group 0. You cannot enable preemption for redundancy group 0.

When you do a manual failover for redundancy group 0, the node in the primary state transitions to the secondary-hold state. The node stays in the secondary-hold state for the default or configured time (a minimum of 300 seconds) and then transitions to the secondary state.

State transitions in cases where one node is in the secondary-hold state and the other node reboots, or the control link connection or fabric link connection is lost to that node, are described as follows:

- Reboot case—The node in the secondary-hold state transitions to the primary state; the other node goes dead (inactive).
- Control link failure case—The node in the secondary-hold state transitions to the ineligible state and then to a disabled state; the other node transitions to the primary state.
- Fabric link failure case—The node in the secondary-hold state transitions directly to the disabled state.

Keep in mind that during an in-service software upgrade (ISSU), the transitions described here cannot happen. Instead, the other (primary) node transitions directly to the secondary state because Juniper releases earlier than 10.0 do not interpret the secondary-hold state. While you start an ISSU, if one of the nodes has one or more redundancy groups in the secondary-hold state, you must wait for them to move to the secondary state before you can do manual failovers to make all the redundancy groups be primary on one node.



**CAUTION:** Be cautious and judicious in your use of redundancy group 0 manual failovers. A redundancy group 0 failover implies a Routing Engine failover, in which case all processes running on the primary node are killed and then spawned on the new master Routing Engine. This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

---

**Related Topics**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Group Failover on page 812
- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 814
- Example: Configuring Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers (CLI) on page 816
- Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 816
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Understanding Chassis Cluster Monitoring of Global-Level Objects on page 810
- Upgrading Both Devices in a Chassis Cluster Using an ISSU on page 874

**Initiating a Chassis Cluster Manual Redundancy Group Failover**

You can initiate a failover manually with the **request** command. A manual failover bumps up the priority of the redundancy group for that member to 255.

Before you begin, complete the following tasks:

- Example: Configuring Chassis Cluster Redundancy Groups (CLI) on page 803
- Example: Configuring Chassis Cluster Redundant Ethernet Interfaces (CLI) on page 819
- Example: Configuring Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers (CLI) on page 816



**CAUTION:** Be cautious and judicious in your use of redundancy group 0 manual failovers. A redundancy group 0 failover implies a Routing Engine (RE) failover, in which case all processes running on the primary node are killed and then spawned on the new master Routing Engine (RE). This failover could result in loss of state, such as routing state, and degrade performance by introducing system churn.

Use the **show** command to display the status of nodes in the cluster:

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 0
Cluster ID: 9
Node Priority Status Preempt Manual failover

Redundancy group: 0 , Failover count: 1
 node0 254 primary no no
 node1 1 secondary no no
```

Output to this command indicates that node 0 is primary.

Use the **request** command to trigger a failover and make node 1 primary:

```
{primary:node0}
user@host> request chassis cluster failover redundancy-group 0 node 1

Initiated manual failover for redundancy group 0
```

Use the **show** command to display the new status of nodes in the cluster:

```
{secondary-hold:node0}
user@host> show chassis cluster status redundancy-group 0
Cluster ID: 9
Node Priority Status Preempt Manual failover

Redundancy group: 0 , Failover count: 2
 node0 254 secondary-hold no yes
 node1 255 primary no yes
```

Output to this command shows that node 1 is now primary and node 0 is in the secondary-hold state. After 5 minutes, node 0 will transition to the secondary state.

You can reset the failover for redundancy groups by using the **request** command. This change is propagated across the cluster.

```
{secondary-hold:node0}
user@host> request chassis cluster failover reset redundancy-group 0 node 0
node0:
```

-----  
No reset required for redundancy group 0.

node1:

-----  
Successfully reset manual failover for redundancy group 0

You cannot trigger a back-to-back failover until the 5-minute interval expires.

```
{secondary-hold:node0}
user@host> request chassis cluster failover redundancy-group 0 node 0
node0:
```

-----  
Manual failover is not permitted as redundancy-group 0 on node0 is in secondary-hold state.

Use the **show** command to display the new status of nodes in the cluster:

```
{secondary-hold:node0}
user@host> show chassis cluster status redundancy-group 0
Cluster ID: 9
Node Priority Status Preempt Manual failover

Redundancy group: 0 , Failover count: 2
 node0 254 secondary-hold no no
 node1 1 primary no no
```

Output to this command shows that a back-to-back failover has not occurred for either node.

After doing a manual failover, you must issue the **reset failover** command before requesting another failover.

When the primary node fails and comes back up, election of the primary node is done based on regular criteria (priority and preempt).

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Group Manual Failover on page 813
- Example: Configuring Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers (CLI) on page 816

- Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 816
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Understanding Chassis Cluster Monitoring of Global-Level Objects on page 810

### Example: Configuring Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers (CLI)

Although back-to-back redundancy group failovers have a default dampening time, you can configure it. For redundancy group 0, the default interval is 5 minutes (300 seconds), with a range of 300 through 1800 seconds. For redundancy group *x* (redundancy groups 1 through 128), the default interval is 1 second, with a range of 0 through 1800 seconds.

Use the following command to set the minimum interval (in seconds) to be allowed between back-to-back failovers for redundancy group 0:

```
{primary:node1}
user@host# set chassis cluster redundancy-group 0 hold-down-interval 420
```

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Group Manual Failover on page 813
- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 814
- Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover on page 816
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Understanding Chassis Cluster Monitoring of Global-Level Objects on page 810

### Understanding SNMP Failover Traps for Chassis Cluster Redundancy Group Failover

Chassis clustering supports SNMP traps, which are triggered whenever there is a redundancy group failover.

The trap message can help you troubleshoot failovers. It contains the following information:

- The cluster ID and node ID
- The reason for the failover
- The redundancy group that is involved in the failover
- The redundancy group's previous state and current state

These are the different states that a cluster can be in at any given instant: hold, primary, secondary-hold, secondary, ineligible, and disabled. Traps are generated for the following state transitions (only a transition from a hold state does not trigger a trap):

- primary <—> secondary
- primary —> secondary-hold



- secondary-hold → secondary
- secondary → ineligible
- ineligible → disabled
- ineligible → primary
- secondary → disabled

A transition can be triggered because of any event, such as interface monitoring, SPU monitoring, failures, and manual failovers.

The trap is forwarded over the control link if the outgoing interface is on a node different from the node on the Routing Engine that generates the trap.

You can specify that a trace log be generated by setting the **traceoptions flag snmp** statement.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundancy Group Manual Failover on page 813
- Initiating a Chassis Cluster Manual Redundancy Group Failover on page 814
- Example: Configuring Chassis Cluster with a Dampening Time Between Back-to-Back Redundancy Group Failovers (CLI) on page 816
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Understanding Chassis Cluster Monitoring of Global-Level Objects on page 810

---

## Chassis Cluster Redundant Ethernet Interfaces

- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Example: Configuring Chassis Cluster Redundant Ethernet Interfaces (CLI) on page 819
- Verifying Chassis Cluster Interfaces on page 821
- Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 822

### Understanding Chassis Cluster Redundant Ethernet Interfaces

A redundant Ethernet interface is a pseudointerface that includes at minimum one physical interface from each node of the cluster.

A redundant Ethernet interface must contain, at minimum, a pair of Fast Ethernet interfaces or a pair of Gigabit Ethernet interfaces that are referred to as child interfaces of the redundant Ethernet interface (the redundant parent). If two or more child interfaces from each node are assigned to the redundant Ethernet interface, a redundant Ethernet interface link aggregation group must be formed. A single redundant Ethernet interface might include a Fast Ethernet interface from node 0 and a Fast Ethernet interface from node 1 or a Gigabit Ethernet interface from node 0 and a Gigabit Ethernet interface from node 1. Although a redundant Ethernet interface's interfaces must be the same kind—either Fast Ethernet or Gigabit Ethernet—they do not need to be in the same slots on each node.

On SRX3400, SRX3600, SRX5600, and SRX5800 devices, 10-Gigabit Ethernet (xe) interfaces can be redundant Ethernet (reth) interfaces.



**NOTE:** A redundant Ethernet interface is referred to as a **reth** in configuration commands.

The maximum number of redundant Ethernet interfaces that you can configure varies, depending on the device type you are using, as shown in Table 78 on page 818. Note that the number of redundant Ethernet interfaces configured determines the number of redundancy groups that can be configured.

**Table 78: Maximum Number of Redundant Ethernet Interfaces Allowed**

| Device  | Maximum Number of reth Interfaces |
|---------|-----------------------------------|
| SRX100  | 8                                 |
| SRX210  | 8                                 |
| SRX240  | 24                                |
| SRX650  | 68                                |
| SRX3400 | 128                               |
| SRX3600 | 128                               |
| SRX5600 | 128                               |
| SRX5800 | 128                               |
| J2320   | 128                               |
| J2350   | 128                               |
| J4350   | 128                               |
| J6350   | 128                               |

A redundant Ethernet interface's child interface is associated with the redundant Ethernet interface as part of the child interface configuration. The redundant Ethernet interface child interface inherits most of its configuration from its parent.



**NOTE:** You can enable promiscuous mode on redundant Ethernet interfaces. When promiscuous mode is enabled on a Layer 3 Ethernet interface, all packets received on the interface are sent to the central point or Services Processing Unit (SPU), regardless of the destination MAC address of the packet. If you enable promiscuous mode on a redundant Ethernet interface, promiscuous mode is then enabled on any child physical interfaces.

To enable promiscuous mode on a redundant Ethernet interface, use the `promiscuous-mode` statement at the `[edit interfaces]` hierarchy.

A redundant Ethernet interface inherits its failover properties from the redundancy group `x` that it belongs to. A redundant Ethernet interface remains active as long as its primary child interface is available or active. For example, if `reth0` is associated with redundancy group 1 and redundancy group 1 is active on node 0, then `reth0` is up as long as the node 0 child of `reth0` is up.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Configuring Chassis Cluster Redundant Ethernet Interfaces (CLI) on page 819
- Verifying Chassis Cluster Interfaces on page 821
- Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 822
- Understanding Conditional Route Advertising in a Chassis Cluster on page 825
- Understanding Chassis Cluster Formation on page 796

### Example: Configuring Chassis Cluster Redundant Ethernet Interfaces (CLI)

A redundant Ethernet interface is a pseudointerface that contains two or more physical interfaces, with at least one from each node of the cluster. Once the physical interfaces have been assigned to the redundant Ethernet interface, you then set the configuration that pertains to them at the level of the redundant Ethernet interface, and each of the child interfaces inherits the configuration.



**NOTE:** A redundant Ethernet interface is referred to as a `reth` in configuration commands.

Before you begin, complete the following tasks:

- Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
- Example: Configuring the Chassis Cluster Fabric (CLI) on page 842
- Example: Configuring Chassis Cluster Redundancy Groups (CLI) on page 803

Use the following commands to bind redundant child physical interfaces to **reth1** and **reth2**, respectively:

```
{primary:node1}
user@host# set interfaces ge-0/0/0 gigether-options redundant-parent reth1
{primary:node1}
user@host# set interfaces ge-7/0/0 gigether-options redundant-parent reth1
{primary:node1}
user@host# set interfaces fe-1/0/0 fast-ether-options redundant-parent reth2
{primary:node1}
user@host# set interfaces fe-8/0/0 fast-ether-options redundant-parent reth2
```

Use the following commands to:

- Add **reth1** to redundancy group 1.
- Set the MTU size to 1500 bytes.
- Assign IP address **10.1.1.3/24** to **reth1**.

```
{primary:node1}
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
{primary:node1}
user@host# set interfaces reth1 unit 0 family inet mtu 1500
{primary:node1}
user@host# set interfaces reth1 unit 0 family inet address 10.1.1.3/24
```

For IPv6:

```
{primary:node1}
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
{primary:node1}
user@host# set interfaces reth1 unit 0 family inet6 mtu 1500
{primary:node1}
user@host# set interfaces reth1 unit 0 family inet6 address 2010:2010:201::2/64
```

Use the following command to associate **reth1.0** with a security zone named **Trust**. Security zone configuration is the same for redundant Ethernet interfaces as for any other interface.

```
{primary:node1}
user@host# set security zones security-zone Trust interfaces reth1.0
```



**NOTE:** You can enable promiscuous mode on redundant Ethernet interfaces. When promiscuous mode is enabled on a Layer 3 Ethernet interface, all packets received on the interface are sent to the central point or Services Processing Unit regardless of the destination MAC address of the packet. If you enable promiscuous mode on a redundant Ethernet interface, promiscuous mode is then enabled on any child physical interfaces. To enable promiscuous mode on a redundant Ethernet interface, use the **promiscuous-mode** statement at the **[edit interfaces]** hierarchy.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817

- Verifying Chassis Cluster Interfaces on page 821
- Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 822
- Understanding Conditional Route Advertising in a Chassis Cluster on page 825
- Understanding Chassis Cluster Formation on page 796

## Verifying Chassis Cluster Interfaces

**Purpose** Display information about chassis cluster interfaces.

**Action** From the CLI, enter the **show chassis cluster interfaces** command:

```
{primary:node1}
user@host> show chassis cluster interfaces
Control link 0 name: em0

Redundant-ethernet Information:
 Name Status Redundancy-group
 reth0 Up 1

Interface Monitoring:
 Interface Weight Status Redundancy-group
 ge-6/0/0 200 Up 1

{primary:node1}
user@host> show chassis cluster interfaces
Control link 0 name: em0
Control link 1 name: em1

Redundant-ethernet Information:
 Name Status Redundancy-group
 reth0 Up 1

Interface Monitoring:
 Interface Weight Status Redundancy-group
 ge-6/0/0 200 Up 1
```



**NOTE:** On SRX3400, SRX3600, SRX5600, and SRX5800 devices, eight-queue configurations are not reflected on the chassis cluster interface.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Configuring Chassis Cluster Redundant Ethernet Interfaces (CLI) on page 819

## Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups

- Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 822
- Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups (CLI) on page 823
- Example: Configuring Chassis Cluster Minimum Links (CLI) on page 824

### Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups

Support for Ethernet link aggregation groups (LAGs) based on IEEE 802.3ad makes it possible to aggregate physical interfaces on a standalone device. LAGs on standalone devices provide increased interface bandwidth and link availability. Aggregation of links in a chassis cluster allows a redundant Ethernet interface to add more than two physical child interfaces thereby creating a redundant Ethernet interface LAG. A redundant Ethernet interface LAG can have up to eight links per redundant Ethernet interface per node (for a total of 16 links per redundant Ethernet interface).

The aggregated links in a redundant Ethernet interface LAG provide the same bandwidth and redundancy benefits of a LAG on a standalone device with the added advantage of chassis cluster redundancy. A redundant Ethernet interface LAG has two types of simultaneous redundancy. The aggregated links within the redundant Ethernet interface on each node are redundant; if one link in the primary aggregate fails, its traffic load is taken up by the remaining links. If enough child links on the primary node fail, the redundant Ethernet interface LAG can be configured so that all traffic on the entire redundant Ethernet interface fails over to the aggregate link on the other node.

Aggregated Ethernet interfaces, known as local LAGs, are also supported on either node of a chassis cluster but cannot be added to redundant Ethernet interfaces. Local LAGs are indicated in the system interfaces list using an `ae-` prefix. Likewise any child interface of an existing local LAG cannot be added to a redundant Ethernet interface and vice versa. Note that it is necessary for the switch (or switches) used to connect the nodes in the cluster to have a LAG link configured and 802.3ad enabled for each LAG on both nodes so that the aggregate links are recognized as such and correctly pass traffic. The total maximum number of combined individual node LAG interfaces (`ae`) and redundant Ethernet (`reth`) interfaces per cluster is 128.



**NOTE:** The redundant Ethernet interface LAG child links from each node in the chassis cluster must be connected to a different LAG at the peer devices. If a single peer switch is used to terminate the redundant Ethernet interface LAG, two separate LAGs must be used in the switch.

---

Links from different PICs or IOCs and using different cable types (for example, copper and fiber-optic) can be added to the same redundant Ethernet interface LAG but the speed of the interfaces must be the same and all interfaces must be in full duplex mode. We recommend, however, that for purposes of reducing traffic processing overhead, interfaces from the same PIC or IOC be used whenever feasible. Regardless, all interfaces configured in a redundant Ethernet interface LAG share the same virtual MAC address.

Redundant Ethernet interface configuration also includes a minimum-links setting that allows you to set a minimum number of physical child links on the primary node in a given redundant Ethernet interface that must be working for the interface to be up. The default minimum-links value is 1. Note that the minimum-links setting only monitors child links on the primary node. Redundant Ethernet interfaces do not use physical interfaces on the backup node for either ingress or egress traffic.

Note the following support details:

- Quality of service (QoS) is supported in a redundant Ethernet interface LAG. Guaranteed bandwidth is, however, duplicated across all links. If a link is lost, there is a corresponding loss of guaranteed bandwidth.
- Layer 2 transparent mode and Layer 2 security features are supported in redundant Ethernet interface LAGs.
- Link Aggregation Control Protocol (LACP) is supported in chassis cluster deployments, where aggregated Ethernet interfaces and redundant Ethernet interfaces are supported simultaneously.
- Chassis cluster management, control, and fabric interfaces cannot be configured as redundant Ethernet interface LAGs or added to a redundant Ethernet interface LAG.
- Network processor bundling can coexist with redundant Ethernet interface LAGs on the same cluster. However, assigning an interface simultaneously to a redundant Ethernet interface LAG and a network processor bundle is not supported.
- Single flow throughput is limited to the speed of a single physical link regardless of the speed of the aggregate interface.



**NOTE:** For more information about Ethernet interface link aggregation and LACP, see the “Aggregated Ethernet” chapter of the *Junos OS Interfaces Configuration Guide for Security Devices*.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups (CLI) on page 823
- Example: Configuring Chassis Cluster Minimum Links (CLI) on page 824
- Understanding Conditional Route Advertising in a Chassis Cluster on page 825
- Understanding Chassis Cluster Formation on page 796

#### Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups (CLI)

Chassis cluster configuration supports more than one child interface per node in a redundant Ethernet interface. When at least two physical child interface links from each node are included in a redundant Ethernet interface configuration, the interfaces are

combined within the redundant Ethernet interface to form a redundant Ethernet interface link aggregation group.



**NOTE:** For the aggregation to take place, the switch used to connect the nodes in the cluster must enable IEEE 802.3ad link aggregation for the redundant Ethernet interface physical child links on each node. Because most switches support IEEE 802.3ad and are also Link Aggregation Control Protocol (LACP) capable, we recommend that you enable LACP on the SRX Series devices. In cases where LACP is not available on the switch, you should not enable LACP on the SRX Series devices.

```
{primary:node1}
user@host# set interfaces ge-1/0/1 gigether-options redundant-parent reth1
{primary:node1}
user@host# set interfaces ge-1/0/2 gigether-options redundant-parent reth1
{primary:node1}
user@host# set interfaces ge-1/0/3 gigether-options redundant-parent reth1
{primary:node1}
user@host# set interfaces ge-12/0/1 gigether-options redundant-parent reth1
{primary:node1}
user@host# set interfaces ge-12/0/2 gigether-options redundant-parent reth1
{primary:node1}
user@host# set interfaces ge-12/0/3 gigether-options redundant-parent reth1
```

All six of these Gigabit Ethernet interfaces, three from each node, have now been assigned to **reth1**. All other steps related to redundant Ethernet interface configuration are the same as those for a redundant Ethernet interface with only two child interfaces.



**NOTE:** A maximum of eight physical interfaces per node in a cluster, for a total of 16 child interfaces, can be assigned to a single redundant Ethernet interface when a redundant Ethernet interface LAG is being configured.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 822
- Example: Configuring Chassis Cluster Minimum Links (CLI) on page 824
- Understanding Conditional Route Advertising in a Chassis Cluster on page 825
- Understanding Chassis Cluster Formation on page 796

#### Example: Configuring Chassis Cluster Minimum Links (CLI)

When a redundant Ethernet interface has more than two child links, you can set a minimum number of physical links assigned to the interface on the primary node that must be working for the interface to be up. When the number of physical links on the primary node falls below the minimum-links value, the interface will be down even if some links are still working. The default minimum-links value is 1.



```
{primary:node1}
user@host# set interfaces reth1 redundant-ether-options minimum-links 3
```

The previous minimum-links setting example requires that three child links on the current primary node and bound to **reth1** be working to prevent the interface from going down. In a redundant Ethernet interface LAG configuration where six interfaces are assigned to **reth1**, setting the minimum-links value to 3 would mean that all **reth1** child links on the primary node must be working to prevent the interface's status from changing to down.



**NOTE:** Although it is possible to set a minimum-links value for a redundant Ethernet interface with only two child interfaces (one on each node), we do not recommend it.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Understanding Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups on page 822
- Example: Configuring Chassis Cluster Redundant Ethernet Interface Link Aggregation Groups (CLI) on page 823
- Understanding Conditional Route Advertising in a Chassis Cluster on page 825
- Understanding Chassis Cluster Formation on page 796

## Conditional Route Advertising in a Chassis Cluster

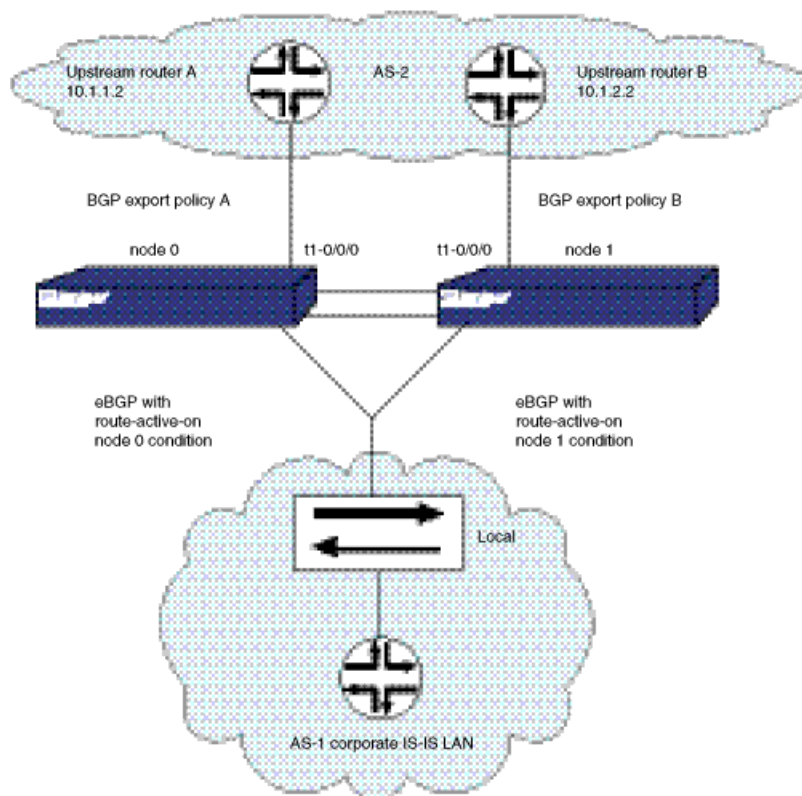
- Understanding Conditional Route Advertising in a Chassis Cluster on page 825
- Example: Configuring Conditional Route Advertising in a Chassis Cluster (CLI) on page 826

### Understanding Conditional Route Advertising in a Chassis Cluster

Route advertisement over redundant Ethernet interfaces in a chassis cluster is complicated by the fact that the active node in the cluster can change dynamically. Conditional route advertisement enables you to advertise routes in such a way that incoming traffic from the core network is attracted to the Border Gateway Protocol (BGP) interface that exists on the same node as the currently active redundant Ethernet interface. In this way, traffic is processed by the active node and does not traverse the fabric interface between nodes. You do this by manipulating the BGP attribute at the time routes are advertised by BGP.

The goal of conditional route advertisement in a chassis cluster is to ensure that incoming traffic from the upstream network arrives on the node that is on the currently active redundant Ethernet interface. To understand how this works, keep in mind that in a chassis cluster, each node has its own set of interfaces. Figure 86 on page 826 shows a typical scenario, with a redundant Ethernet interface connecting the corporate LAN, through a chassis cluster, to an external network segment.

Figure 86: Conditional Route Advertising



- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
  - Example: Configuring Conditional Route Advertising in a Chassis Cluster (CLI) on page 826
  - Understanding Chassis Cluster Formation on page 796

### Example: Configuring Conditional Route Advertising in a Chassis Cluster (CLI)

As illustrated in Figure 86 on page 826, routing prefixes learned from the redundant Ethernet interface through the IGP are advertised toward the network core using BGP. Two BGP sessions are maintained, one from interface **t1-1/0/0** and one from **t1-1/0/1** for BGP multihoming. All routing prefixes are advertised on both sessions. Thus, for a route advertised by BGP, learned over a redundant Ethernet interface, if the active redundant Ethernet interface is on the same node as the BGP session, you advertise the route with a “good” BGP attribute.

To achieve this behavior, you apply a policy to BGP before exporting routes. An additional term in the policy match condition determines the current active redundant Ethernet interface child interface of the next hop before making the routing decision. When the active status of a child redundant Ethernet interface changes, BGP reevaluates the export policy for all routes affected.

```
{primary:node1}
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0
 from protocol ospf
{primary:node1}
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0
 from condition reth-nh-active-on-0
{primary:node1}
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0
 then metric 10
{primary:node1}
user@host# set policy-options policy-statement reth-nh-active-on-0 term ospf-on-0
 then accept
{primary:node1}
user@host# set policy-options condition reth-nh-active-on-0 route-active-on node0
```

The condition statement in this configuration works as follows. The command states that any routes evaluated against this condition will pass only if:

- The routes have a redundant Ethernet interface as their next-hop interface.
- The current child interface of the redundant Ethernet interface in the previous example is active at node 0 (as specified by the **route-active-on node0** keyword).

```
{primary:node1}
user@host# set policy-options condition reth-nh-active-on-0 route-active-on node0
```

Note that a route might have multiple equal-cost next hops, and those next hops might be redundant Ethernet interfaces, regular interfaces, or a combination of both. The route still satisfies the requirement that it has a redundant Ethernet interface as its next hop.

If you use the BGP export policy set for node 0 in the previous example command, only OSPF routes that satisfy the following requirements will be advertised through the session:

- The OSPF routes have a redundant Ethernet interface as their next hop.
- The current child interface of the redundant Ethernet interface is currently active at node 0.

You must also create and apply a separate policy statement for the other BGP session by using this same process.

In addition to the BGP MED attribute, you can define additional BGP attributes, such as **origin-code**, **as-path**, and **community**.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Redundant Ethernet Interfaces on page 817
- Understanding Conditional Route Advertising in a Chassis Cluster on page 825

- [Understanding Chassis Cluster Formation on page 796](#)

## Chassis Cluster Control Plane

---

- [Understanding the Chassis Cluster Control Plane on page 828](#)
- [Understanding Chassis Cluster Control Links on page 829](#)
- [Example: Configuring Chassis Cluster Control Ports \(CLI\) on page 830](#)
- [Understanding Chassis Cluster Dual Control Links on page 830](#)
- [Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 832](#)
- [Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices on page 833](#)
- [Understanding Chassis Cluster Control Link Heartbeats on page 834](#)
- [Understanding Chassis Cluster Control Link Failure and Recovery on page 835](#)
- [Example: Configuring Chassis Cluster Control Link Recovery \(CLI\) on page 837](#)
- [Verifying Chassis Cluster Control Plane Statistics on page 837](#)
- [Clearing Chassis Cluster Control Plane Statistics on page 838](#)

## Understanding the Chassis Cluster Control Plane

The control plane software, which operates in active or backup mode, is an integral part of Junos OS that is active on the primary node of a cluster. It achieves redundancy by communicating state, configuration, and other information to the inactive Routing Engine on the secondary node. If the master Routing Engine fails, the secondary one is ready to assume control.

The control plane software:

- Runs on the Routing Engine and oversees the entire chassis cluster system, including interfaces on both nodes
- Manages system and data plane resources, including the Packet Forwarding Engine (PFE) on each node
- Synchronizes the configuration over the control link
- Establishes and maintains sessions, including authentication, authorization, and accounting (AAA) functions
- Manages application-specific signaling protocols
- Establishes and maintains management sessions, such as Telnet connections
- Handles asymmetric routing
- Manages routing state, Address Resolution Protocol (ARP) processing, and Dynamic Host Configuration Protocol (DHCP) processing

Information from the control plane software follows two paths:

- On the primary node (where the Routing Engine is active), control information flows from the Routing Engine to the local Packet Forwarding Engine.
- Control information flows across the control link to the secondary node's Routing Engine and Packet Forwarding Engine.

The control plane software running on the master Routing Engine maintains state for the entire cluster, and only processes running on its node can update state information. The master Routing Engine synchronizes state for the secondary node and also processes all host traffic.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Chassis Cluster Control Links on page 829
  - Example: Configuring Chassis Cluster Control Ports (CLI) on page 830
  - Understanding Chassis Cluster Dual Control Links on page 830
  - Understanding Chassis Cluster Control Link Heartbeats on page 834
  - Understanding Chassis Cluster Control Link Failure and Recovery on page 835
  - Understanding Chassis Cluster Formation on page 796

## Understanding Chassis Cluster Control Links

The control link relies on a proprietary protocol to transmit session state, configuration, and liveness signals across the nodes.

On SRX5600 and SRX5800 devices, by default, all control ports are disabled. Each SPC in a device has two control ports, and each device can have multiple SPCs plugged into it. To set up the control link in a chassis cluster with SRX5600 or SRX5800 devices, you connect and configure the control ports that you will use on each device (**fpcn** and **fpcn**) and then initialize the device in cluster mode.

For SRX3400 and SRX3600 devices, there are dedicated chassis cluster (HA) control ports on the switch fabric board. No control link configuration is needed for SRX3400 and SRX3600 devices.

For SRX650 and SRX240 devices, the control link uses the **ge-0/0/1** interface.

For SRX210 and SRX100 devices, the control link uses the **fe-0/0/7** interface.

In a J Series chassis cluster, the control link is a physical connection between the **ge-0/0/3** ports on each device, with both transformed into **fxp1**.

For details about port and interface usage for management, control, and fabric links, see Table 80 on page 848 and Table 81 on page 854.

To set up the control link on J Series devices, you connect the control interfaces on the two devices back-to-back. When you initialize a device in cluster mode, Junos OS renames the control interface to **fxp1** and uses that interface for the cluster control link. To enable the control link to transmit data, the system provides each **fxp1** control link interface with an internal IP address.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Configuring Chassis Cluster Control Ports (CLI) on page 830
  - Understanding Chassis Cluster Dual Control Links on page 830
  - Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 832
  - Understanding Chassis Cluster Control Link Heartbeats on page 834
  - Understanding Chassis Cluster Control Link Failure and Recovery on page 835
  - Understanding the Chassis Cluster Control Plane on page 828

### Example: Configuring Chassis Cluster Control Ports (CLI)

On SRX5600 and SRX5800 devices, by default, all control ports are disabled. You need to configure the control ports that you will use on each device to set up the control link or, optionally, the dual control links in a chassis cluster.

Use the following commands to configure the control ports for use as the control link for the chassis cluster:

```
user@host# set chassis cluster control-ports fpc4 port 0
user@host# set chassis cluster control-ports fpc10 port 0
```

Use the following commands to configure the control ports for use as dual control links for the chassis cluster.

```
user@host# set chassis cluster control-ports fpc4 port 0
user@host# set chassis cluster control-ports fpc10 port 0
user@host# set chassis cluster control-ports fpc6 port 1
user@host# set chassis cluster control-ports fpc12 port 1
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Chassis Cluster Control Links on page 829
  - Understanding Chassis Cluster Dual Control Links on page 830
  - Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 832
  - Understanding Chassis Cluster Control Link Heartbeats on page 834
  - Understanding Chassis Cluster Control Link Failure and Recovery on page 835
  - Understanding the Chassis Cluster Control Plane on page 828

### Understanding Chassis Cluster Dual Control Links

Dual control links, where two pairs of control link interfaces are connected between each device in a cluster, are supported for the SRX5000 and SRX3000 lines. Having two control links helps to avoid a possible single point of failure.

For the SRX5000 line, this functionality requires a second Routing Engine, as well as a second Switch Control Board (SCB) to house the Routing Engine, to be installed on each

device in the cluster. The purpose of the second Routing Engine is only to initialize the switch on the SCB.

For the SRX3000 line, this functionality requires an SRX Clustering Module (SCM) to be installed on each device in the cluster. Although the SCM fits in the Routing Engine slot, it is not a Routing Engine. SRX3000 line devices do not support a second Routing Engine. The purpose of the SCM is to initialize the second control link.



**NOTE:** For the SRX5000 line, the second Routing Engine must be running Junos OS Release 10.0 or later. For the SRX3000 line, the cluster must be running Junos OS Release 10.2 or later (the SCM is not supported in earlier releases and might be incorrectly recognized).

The second Routing Engine, to be installed on SRX5000 line devices only, does not provide backup functionality. It does not need to be upgraded, even when there is a software upgrade of the master Routing Engine on the same node. Note the following conditions:

- You cannot run the CLI or enter configuration mode on the second Routing Engine.
- You do not need to set the chassis ID and cluster ID on the second Routing Engine.
- You need only a console connection to the second Routing Engine. (A console connection is not needed unless you want to check that the second Routing Engine booted up or to upgrade a software image.)
- You cannot log in to the second Routing Engine from the master Routing Engine.



**NOTE:** As long as the first Routing Engine is installed (even if it is rebooting or failing), the second Routing Engine cannot take over the chassis mastership; that is, it cannot control all the hardware on the chassis. The second Routing Engine can only become the master when the master Routing Engine is not present.

#### Related Topics

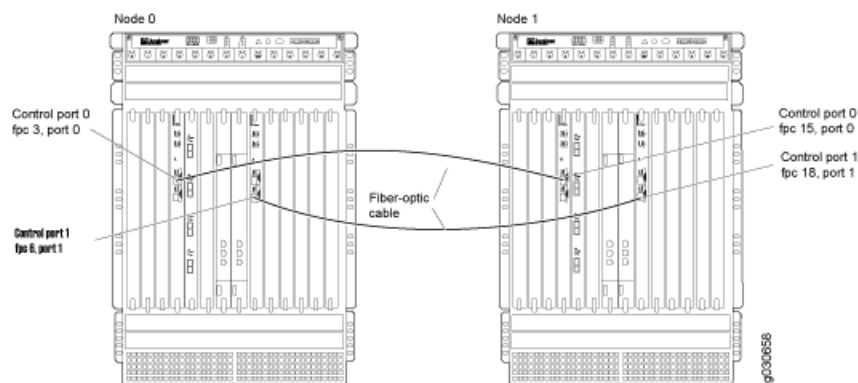
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Control Links on page 829
- Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 832
- Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices on page 833
- Understanding Chassis Cluster Control Link Heartbeats on page 834
- Understanding Chassis Cluster Control Link Failure and Recovery on page 835
- Understanding the Chassis Cluster Control Plane on page 828

## Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster

For the SRX5000 and SRX3000 lines, you can connect two control links between the two devices, effectively reducing the chance of control link failure.

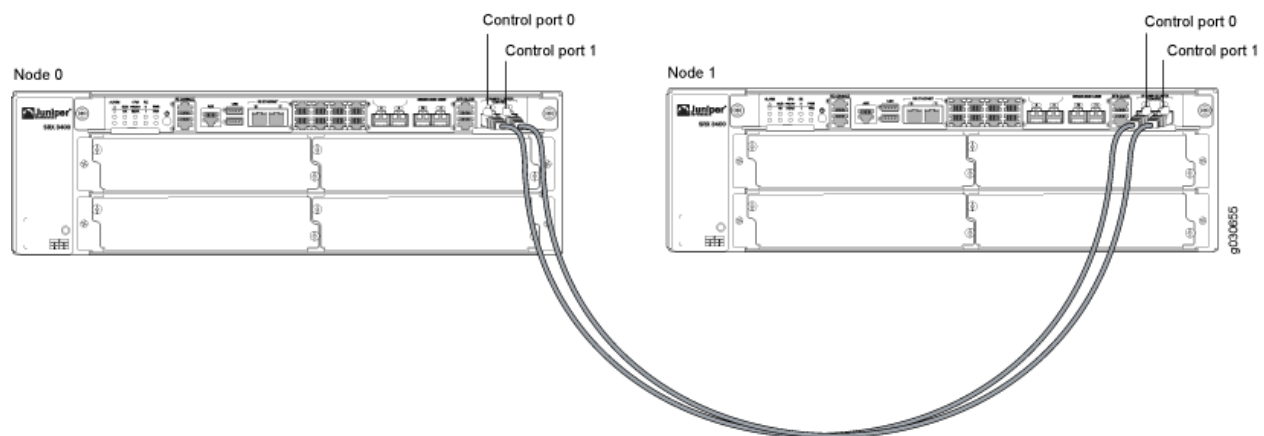
For devices in the SRX5000 line, connect two pairs of the same type of Ethernet ports. For each device, you can use ports on the same Services Processing Card (SPC), but we recommend that they be on two different SPCs to provide high availability. Figure 87 on page 832 shows a pair of SRX5800 devices with dual control links connected. In this example, control port 0 and control port 1 are connected on different SPCs.

**Figure 87: Connecting Dual Control Links (SRX5800 Devices)**



For devices in the SRX3000 line, connect two pairs of the same type of Ethernet ports. For each device, use both available built-in ports. Figure 88 on page 832 shows a pair of SRX3400 devices with dual control links connected.

**Figure 88: Connecting Dual Control Links (SRX3400 Devices)**



**NOTE:** For devices in both the SRX5000 and SRX3000 lines, you must connect control port 0 on one node to control port 0 on the other node and, likewise, control port 1 to control port 1. If you connect control port 0 to control port 1, the nodes cannot receive heartbeat packets across the control links.



- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Chassis Cluster Control Links on page 829
  - Understanding Chassis Cluster Dual Control Links on page 830
  - Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices on page 833
  - Understanding Chassis Cluster Control Link Heartbeats on page 834
  - Understanding Chassis Cluster Control Link Failure and Recovery on page 835
  - Understanding the Chassis Cluster Control Plane on page 828

## Upgrading the Second Routing Engine When Using Chassis Cluster Dual Control Links on SRX5600 and SRX5800 Devices

For SRX5600 and SRX5800 devices, a second Routing Engine is required for each device in a cluster if you are using dual control links. The second Routing Engine does not provide backup functionality; its purpose is only to initialize the switch on the Switch Control Board (SCB). The second Routing Engine must be running Junos OS Release 10.0 or later.

Because you cannot run the CLI or enter configuration mode on the second Routing Engine, you cannot upgrade the Junos OS image with the usual upgrade commands. Instead, use the master Routing Engine (RE0) to create a bootable USB storage device, which you can then use to install a software image on the second Routing Engine (RE1).

To upgrade the software image on the second Routing Engine (RE1):

1. Use FTP to copy the installation media into the **/var/tmp** directory of the master Routing Engine (RE0).
2. Insert a USB storage device into the USB port on the master Routing Engine (RE0).
3. In the UNIX shell, navigate to the **/var/tmp** directory:

```
start shell
cd /var/tmp
```

4. Log in as root or superuser:

```
su [enter]
password: [enter SU password]
```

5. Issue the following command:

```
dd if=installMedia of=/dev/externalDrive bs=64
```

where

- **externalDrive**—Refers to the removable media name. For example, the removable media name on an SRX5000 line device is **da0** for both Routing Engines.
- **installMedia**—Refers to the installation media downloaded into the **/var/tmp** directory. For example, **install-media-srx5000-10.1R1-domestic.tgz**.

The following code example can be used to write the image that you copied to the master Routing Engine (RE0) in step 1 onto the USB storage device:

```
dd if=install-media-srx5000-10.1R1-domestic.tgz of=/dev/da0 bs=64k
```

6. Log out as root or superuser:

```
exit
```

7. After the software image is written to the USB storage device, remove the device and insert it into the USB port on the second Routing Engine (RE1).
8. Move the console connection from the master Routing Engine (RE0) to the second Routing Engine (RE1), if you do not already have a connection.
9. Reboot the second Routing Engine (RE1). Issue the following command:

```
reboot
```

- When the following system output appears, press **y**:

```
WARNING: The installation will erase the contents of your disks.
Do you wish to continue (y/n)?
```

- When the following system output appears, remove the USB storage device and press **Enter**:

```
Eject the installation media and hit [Enter] to reboot?
```

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Control Links on page 829
- Understanding Chassis Cluster Dual Control Links on page 830
- Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 832
- Understanding Chassis Cluster Control Link Heartbeats on page 834
- Understanding Chassis Cluster Control Link Failure and Recovery on page 835
- Understanding the Chassis Cluster Control Plane on page 828

## Understanding Chassis Cluster Control Link Heartbeats

Junos OS transmits heartbeat signals over the control link at a configured interval. The system uses heartbeat transmissions to determine the “health” of the control link. If the number of missed heartbeats has reached the configured threshold, the system assesses whether a failure condition exists.

You specify the heartbeat threshold and heartbeat interval when you configure the chassis cluster.

The system monitors the control link's status by default.

For dual control links, which are supported on the SRX5000 and SRX3000 lines, the Juniper Services Redundancy Protocol process (jsrpd) sends and receives the control

heartbeat messages on both control links. As long as heartbeats are received on one of the control links, Junos OS considers the other node to be alive.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Chassis Cluster Control Links on page 829
  - Understanding Chassis Cluster Dual Control Links on page 830
  - Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 832
  - Understanding Chassis Cluster Control Link Failure and Recovery on page 835
  - Understanding the Chassis Cluster Control Plane on page 828

## Understanding Chassis Cluster Control Link Failure and Recovery

If the control link fails, Junos OS disables the secondary node to prevent the possibility of each node becoming primary for all redundancy groups, including redundancy group 0.

A control link failure is described as not receiving heartbeats over the control link; however, heartbeats are still received over the fabric link.

In the event of a legitimate control link failure, redundancy group 0 remains primary on the node on which it is currently primary, inactive redundancy groups x on the primary node become active, and the secondary node enters a disabled state.



**NOTE:** When the secondary node is disabled, you can still log in to the management port and run diagnostics.

To determine if a legitimate control link failure has occurred, the system relies on redundant liveliness signals sent across the control link and the data link.

The system periodically transmits probes over the fabric data link and heartbeat signals over the control link. Probes and heartbeat signals share a common sequence number that maps them to a unique time event. The software identifies a legitimate control link failure if the following two conditions exist:

- The threshold number of heartbeats were lost.
- At least one probe with a sequence number corresponding to that of a missing heartbeat signal was received on the data link.

When a legitimate control link failure occurs, the following conditions apply:

- Redundancy group 0 remains primary on the node on which it is presently primary (and thus its Routing Engine remains active), and all redundancy groups x on the node become primary.

If the system cannot determine which Routing Engine is primary, the node with the higher priority value for redundancy group 0 is primary and its Routing Engine is active.

(You configure the priority for each node when you configure the **redundancy-group** statement for redundancy group 0.)

- The system disables the secondary node.

To recover a device from the disabled mode, you must reboot the device. When you reboot the disabled node, the node synchronizes its dynamic state with the primary node.



**NOTE:** If you make any changes to the configuration while the secondary node is disabled, execute the **commit** command to synchronize the configuration after you reboot the node. If you did not make configuration changes, the configuration file remains synchronized with that of the primary node.

---

You cannot enable preemption for redundancy group 0. If you want to change the primary node for redundancy group 0, you must do a manual failover.

When you use dual control links (supported on the SRX5000 and SRX3000 lines), note the following conditions:

- Host inbound or outbound traffic can be impacted for up to 3 seconds during a control link failure. For example, consider a case where redundancy group 0 is primary on node 0 and there is a Telnet session to the Routing Engine through a network interface port on node 1. If the currently active control link fails, the Telnet session will lose packets for 3 seconds, until this failure is detected.
- A control link failure that occurs while the commit process is running across two nodes might lead to commit failure. In this situation, run the **commit** command again after 3 seconds.



**NOTE:** Dual control links require a second Routing Engine on each node of the chassis cluster.

---

You can specify that control link recovery be done automatically by the system by setting the **control-link-recovery** statement. In this case, once the system determines that the control link is healthy, it issues an automatic reboot on the disabled node. When the disabled node reboots, the node joins the cluster again.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Control Links on page 829
- Understanding Chassis Cluster Dual Control Links on page 830
- Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 832
- Understanding Chassis Cluster Control Link Heartbeats on page 834
- Example: Configuring Chassis Cluster Control Link Recovery (CLI) on page 837
- Verifying Chassis Cluster Control Plane Statistics on page 837

## Example: Configuring Chassis Cluster Control Link Recovery (CLI)

Although control link recovery is disabled by default, you can configure it.

Use the following command to configure control link recovery:

```
{primary:node1}
user@host# set chassis cluster control-link-recovery
```

Control link recovery is done automatically by the system. After the control link recovers, the system takes the following actions:

- It checks whether it receives at least 30 consecutive heartbeats on the control link, or in the case of dual control links (SRX5000 and SRX3000 lines only), on either control link. This is to ensure that the control link is not flapping and is healthy.
- After it determines that the control link is healthy, the system issues an automatic reboot on the node that was disabled when the control link failed. When the disabled node reboots, it can rejoin the cluster. There is no need for any manual intervention.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Chassis Cluster Control Links on page 829
  - Understanding Chassis Cluster Dual Control Links on page 830
  - Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 832
  - Understanding Chassis Cluster Control Link Heartbeats on page 834
  - Understanding Chassis Cluster Control Link Failure and Recovery on page 835
  - Verifying Chassis Cluster Control Plane Statistics on page 837

## Verifying Chassis Cluster Control Plane Statistics

**Purpose** Display chassis cluster control-plane statistics.

**Action** From the CLI, enter the **show chassis cluster control-plane statistics** command:

```
{primary:node1}
user@host> show chassis cluster control-plane statistics
```

```
Control link statistics:
 Control link 0:
 Heartbeat packets sent: 124
 Heartbeat packets received: 125
```

```
Fabric link statistics:
 Probes sent: 124
 Probes received: 125
```

```
{primary:node1}
user@host> show chassis cluster control-plane statistics
```

```
Control link statistics:
 Control link 0:
 Heartbeat packets sent: 258698
 Heartbeat packets received: 258693
```

```
Control link 1:
 Heartbeat packets sent: 258698
 Heartbeat packets received: 258693
Fabric link statistics:
 Probes sent: 258690
 Probes received: 258690
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster on page 832
  - Example: Configuring Chassis Cluster Control Link Recovery (CLI) on page 837
  - Clearing Chassis Cluster Control Plane Statistics on page 838

## Clearing Chassis Cluster Control Plane Statistics

**Purpose** Clear displayed chassis cluster control plane statistics.

**Action** From the CLI, enter the **clear chassis cluster control-plane statistics** command:

```
{primary:node1}
user@host> clear chassis cluster control-plane statistics

Cleared control-plane statistics
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Configuring Chassis Cluster Control Link Recovery (CLI) on page 837
  - Verifying Chassis Cluster Control Plane Statistics on page 837

## Chassis Cluster Data Plane

---

- Understanding the Chassis Cluster Data Plane on page 838
- Understanding Chassis Cluster Fabric Links on page 840
- Understanding Chassis Cluster Dual Fabric Links on page 841
- Example: Configuring the Chassis Cluster Fabric (CLI) on page 842
- Verifying Chassis Cluster Data Plane Interfaces on page 844
- Verifying Chassis Cluster Data Plane Statistics on page 844
- Clearing Chassis Cluster Data Plane Statistics on page 845

## Understanding the Chassis Cluster Data Plane

The data plane software, which operates in active/active mode, manages flow processing and session state redundancy and processes transit traffic. All packets belonging to a particular session are processed on the same node to ensure that the same security treatment is applied to them. The system identifies the node on which a session is active and forwards its packets to that node for processing. (After a packet is processed, the Packet Forwarding Engine transmits the packet to the node on which its egress interface exists if that node is not the local one.)

To provide for session (or flow) redundancy, the data plane software synchronizes its state by sending special payload packets called runtime objects (RTOs) from one node to the other across the fabric data link. By transmitting information about a session between the nodes, RTOs ensure the consistency and stability of sessions if a failover were to occur, and thus they enable the system to continue to process traffic belonging to existing sessions. To ensure that session information is always synchronized between the two nodes, the data plane software gives RTOs transmission priority over transit traffic.

- Understanding Session RTOs on page 839
- Understanding Data Forwarding on page 839
- Understanding Fabric Data Link Failure and Recovery on page 840

## Understanding Session RTOs

The data plane software creates RTOs for UDP and TCP sessions and tracks state changes. It also synchronizes traffic for IPv4 pass-through protocols such as Generic Routing Encapsulation (GRE) and IPsec.

RTOs for synchronizing a session include:

- Session creation RTOs on the first packet
- Session deletion and ageout RTOs
- Change-related RTOs, including:
  - TCP state changes
  - Timeout synchronization request and response messages
- RTOs for creating and deleting temporary openings in the firewall (pinholes) and child session pinholes

## Understanding Data Forwarding

For Junos OS, flow processing occurs on a single node on which the session for that flow was established and is active. This approach ensures that the same security measures are applied to all packets belonging to a session.

A chassis cluster can receive traffic on an interface on one node and send it out an interface on the other node. (In active/active mode, the ingress interface for traffic might exist on one node and its egress interface on the other.)

This traversal is required in the following situations:

- When packets are processed on one node, but need to be forwarded out an egress interface on the other node
- When packets arrive on an interface on one node, but must be processed on the other node

If the ingress and egress interfaces for a packet are on one node, but the packet must be processed on the other node because its session was established there, it must

traverse the data link twice. This can be the case for some complex media sessions, such as voice-over-IP (VoIP) sessions.

## Understanding Fabric Data Link Failure and Recovery

---



**NOTE:** Intrusion Detection and Prevention (IDP) services do not support failover. For this reason, IDP services are not applied for sessions that were present prior to the failover. IDP services are applied for new sessions created on the new primary node.

---

The fabric data link is vital to the chassis cluster. If the link is unavailable, traffic forwarding and RTO synchronization are affected, which can result in loss of traffic and unpredictable system behavior.

To eliminate this possibility, Junos OS detects fabric faults and disables one node of the cluster. It determines that a fabric fault has occurred if a fabric probe is not received but the fabric interface is active.

To recover from this state, you must reboot the disabled node. When you reboot it, the node synchronizes its state and RTOs with the primary node.

---



**NOTE:** If you make any changes to the configuration while the secondary node is disabled, execute the `commit` command to synchronize the configuration after you reboot the node. If you did not make configuration changes, the configuration file remains synchronized with that of the primary node.

---

### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Chassis Cluster Dual Fabric Links on page 841
- Example: Configuring the Chassis Cluster Fabric (CLI) on page 842
- Verifying Chassis Cluster Data Plane Interfaces on page 844
- Verifying Chassis Cluster Data Plane Statistics on page 844
- Clearing Chassis Cluster Data Plane Statistics on page 845
- Understanding Chassis Cluster Formation on page 796

## Understanding Chassis Cluster Fabric Links

The data link is referred to as the fabric interface. It is used by the cluster's Packet Forwarding Engines to transmit transit traffic and to synchronize the data plane software's dynamic runtime state. When the system creates the fabric interface, the software assigns it an internally derived IP address to be used for packet transmission.

The fabric is a physical connection between two nodes of a cluster and is formed by connecting a pair of Ethernet interfaces back-to-back (one from each node).

Unlike for the control link, whose interfaces are determined by the system, you specify the physical interfaces to be used for the fabric data link in the configuration.



For SRX Series chassis clusters, the fabric link can be any pair of Ethernet interfaces spanning the cluster; for J Series chassis clusters, the fabric link can be any pair of Gigabit Ethernet interface.



**NOTE:** For SRX Series chassis clusters made up of SRX100, SRX210, SRX240, or SRX650, SFP interfaces on mini-PIMs cannot be used as the fabric link.

Table 79 on page 841 shows the fabric interface types that are supported for SRX Series devices.

**Table 79: Supported Fabric Interface Types for SRX Series Devices**

| SRX5000 line        | SRX3000 line        | SRX650           | SRX240           | SRX210           | SRX100        |
|---------------------|---------------------|------------------|------------------|------------------|---------------|
| Fast Ethernet       | Fast Ethernet       | Fast Ethernet    | Fast Ethernet    | Fast Ethernet    | Fast Ethernet |
| Gigabit Ethernet    | Gigabit Ethernet    | Gigabit Ethernet | Gigabit Ethernet | Gigabit Ethernet |               |
| 10-Gigabit Ethernet | 10-Gigabit Ethernet |                  |                  |                  |               |

For details about port and interface usage for management, control, and fabric links, see Table 80 on page 848 and Table 81 on page 854.

The fabric data link does not support fragmentation. To accommodate this state, jumbo frame support is enabled by default on the link with an MTU size of 8940 bytes. To ensure that traffic that transits the data link does not exceed this size, we recommend that no other interfaces exceed the fabric data link's MTU size.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding the Chassis Cluster Data Plane on page 838
- Understanding Chassis Cluster Dual Fabric Links on page 841
- Example: Configuring the Chassis Cluster Fabric (CLI) on page 842
- Verifying Chassis Cluster Data Plane Interfaces on page 844
- Verifying Chassis Cluster Data Plane Statistics on page 844
- Clearing Chassis Cluster Data Plane Statistics on page 845
- Understanding Chassis Cluster Formation on page 796

## Understanding Chassis Cluster Dual Fabric Links

You can connect two fabric links between each device in a cluster, which provides a redundant fabric link between the members of a cluster. Having two fabric links helps to avoid a possible single point of failure.

When you use dual fabric links, the RTOs and probes are sent on one link and the fabric-forwarded and flow-forwarded packets are sent on the other link. If one fabric link

fails, the other fabric link handles the RTOs and probes, as well as the data forwarding. The system selects the physical interface with the lowest slot, PIC, or port number on each node for the RTOs and probes.

For all SRX Series and J Series devices, you can connect two fabric links between the two devices, effectively reducing the chance of control link failure.

In most SRX Series devices in a chassis cluster, you can configure any pair of Gigabit Ethernet interfaces or any pair of 10-Gigabit interfaces to serve as the fabric between nodes. Interfaces on SRX210 devices are Fast Ethernet or Gigabit Ethernet (the paired interfaces must be of a similar type) and all interfaces on SRX100 devices are Fast Ethernet interfaces.

In a J Series chassis cluster, you can configure any pair of Gigabit Ethernet interfaces to serve as the fabric between nodes.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding the Chassis Cluster Data Plane on page 838
  - Understanding Chassis Cluster Fabric Links on page 840
  - Example: Configuring the Chassis Cluster Fabric (CLI) on page 842
  - Verifying Chassis Cluster Data Plane Interfaces on page 844
  - Verifying Chassis Cluster Data Plane Statistics on page 844
  - Clearing Chassis Cluster Data Plane Statistics on page 845
  - Understanding Chassis Cluster Formation on page 796

### Example: Configuring the Chassis Cluster Fabric (CLI)

The fabric is the back-to-back data connection between the nodes in a cluster. Traffic on one node that needs to be processed on the other node or to exit through an interface on the other node passes over the fabric. Session state information also passes over the fabric.

Before you begin, set the chassis cluster node ID and cluster ID using the instructions in “Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI)” on page 866

In most SRX Series devices in a chassis cluster, you can configure any pair of Gigabit Ethernet interfaces or any pair of 10-Gigabit interfaces to serve as the fabric between nodes. Interfaces on SRX210 devices are Fast Ethernet or Gigabit Ethernet (the paired interfaces must be of a similar type) and all interfaces on SRX100 devices are Fast Ethernet interfaces.

In a J Series chassis cluster, you can configure any pair of Gigabit Ethernet interfaces to serve as the fabric between nodes.

You cannot configure filters, policies, or services on the fabric interface. Fragmentation is not supported on the fabric link. The maximum transmission unit (MTU) size is 8,980 bytes. We recommend that no interface in the cluster exceed this MTU. Jumbo frame support on the member links is enabled by default.

Enter the following commands to join **ge-0/0/1** on one node in the cluster and **ge-7/0/1** on the other to form the fabric:

```
{primary:node0}
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/1
{secondary:node1}
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/1
```

For dual fabric links, enter the following commands to join **ge-3/0/0** on one node in the cluster and **ge-10/0/0** on the other to form one fabric and to join **ge-0/0/0** and **ge-7/0/0** to form the second fabric in the cluster:

```
{primary:node0}
user@host# set interfaces fab0 fabric-options member-interfaces ge-3/0/0
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/0
{secondary:node1}
user@host# set interfaces fab1 fabric-options member-interfaces ge-10/0/0
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/0
```

This example is a typical configuration where the dual fabric links are formed with matching slots/ports on each node. That is, **ge-3/0/0** on node0 and **ge-10/0/0** on node1 match, as do **ge-0/0/0** on node0 and **ge-7/0/0** on node1 (the FPC slot offset is 7).

If you choose to configure a different slot/port for the other node, as shown in the following example, make sure you physically connect the RTO-and-probes link to the RTO-and-probes link on the other node. Likewise, make sure you physically connect the data link to the data link on the other node.

```
{primary:node0}
user@host# set interfaces fab0 fabric-options member-interfaces ge-2/1/9
user@host# set interfaces fab0 fabric-options member-interfaces ge-2/2/5
{secondary:node1}
user@host# set interfaces fab1 fabric-options member-interfaces ge-11/0/0
user@host# set interfaces fab1 fabric-options member-interfaces ge-11/3/0
```

That is, physically connect the following two pairs:

- The node0 RTO-and-probes link **ge-2/1/9** to the node1 RTO-and-probes link **ge-11/0/0**
- The node0 data link **ge-2/2/5** to the node1 data link **ge-11/3/0**

Only the same type of interfaces can be configured as fabric children, and you must configure an equal number of child links for **fab0** and **fab1**.



**NOTE:** If you are connecting each of the fabric links through a switch, you must enable the jumbo frame feature on the corresponding switch ports. If both of the fabric links are connected through the same switch, the RTO-and-probes pair must be in one virtual LAN (VLAN) and the data pair must be in another VLAN. Here too, the jumbo frame feature must be enabled on the corresponding switch ports.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding the Chassis Cluster Data Plane on page 838

- Understanding Chassis Cluster Fabric Links on page 840
- Understanding Chassis Cluster Dual Fabric Links on page 841
- Connecting Dual Fabric Links for Devices in a Chassis Cluster
- Verifying Chassis Cluster Data Plane Interfaces on page 844
- Verifying Chassis Cluster Data Plane Statistics on page 844
- Clearing Chassis Cluster Data Plane Statistics on page 845
- Understanding Chassis Cluster Formation on page 796

## Verifying Chassis Cluster Data Plane Interfaces

**Purpose** Display chassis cluster data plane interface status.

**Action** From the CLI, enter the **show chassis cluster data-plane interfaces** command:

```
{primary:node1}
user@host> show chassis cluster data-plane interfaces
fab0:
 Name Status
 ge-2/1/9 up
 ge-2/2/5 up
fab1:
 Name Status
 ge-8/1/9 up
 ge-8/2/5 up
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Configuring the Chassis Cluster Fabric (CLI) on page 842
  - Verifying Chassis Cluster Data Plane Statistics on page 844
  - Clearing Chassis Cluster Data Plane Statistics on page 845

## Verifying Chassis Cluster Data Plane Statistics

**Purpose** Display chassis cluster data plane statistics.

**Action** From the CLI, enter the **show chassis cluster data-plane statistics** command:

```
{primary:node1}
user@host> show chassis cluster data-plane statistics

Services Synchronized:
 Service name RTOs sent RTOs received
 Translation context 0 0
 Incoming NAT 0 0
 Resource manager 0 0
 Session create 0 0
 Session close 0 0
 Session change 0 0
 Gate create 0 0
 Session ageout refresh requests 0 0
 Session ageout refresh replies 0 0
```

|                              |   |   |
|------------------------------|---|---|
| IPSec VPN                    | 0 | 0 |
| Firewall user authentication | 0 | 0 |
| MGCP ALG                     | 0 | 0 |
| H323 ALG                     | 0 | 0 |
| SIP ALG                      | 0 | 0 |
| SCCP ALG                     | 0 | 0 |
| PPTP ALG                     | 0 | 0 |
| RTSP ALG                     | 0 | 0 |

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Configuring the Chassis Cluster Fabric (CLI) on page 842
  - Verifying Chassis Cluster Data Plane Interfaces on page 844
  - Clearing Chassis Cluster Data Plane Statistics on page 845

## Clearing Chassis Cluster Data Plane Statistics

**Purpose** Clear displayed chassis cluster data plane statistics.

**Action** From the CLI, enter the **clear chassis cluster data-plane statistics** command:

```
{primary:node1}
user@host> clear chassis cluster data-plane statistics
```

```
Cleared data-plane statistics
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Configuring the Chassis Cluster Fabric (CLI) on page 842
  - Verifying Chassis Cluster Data Plane Interfaces on page 844
  - Verifying Chassis Cluster Data Plane Statistics on page 844

## Consequences of Enabling Chassis Cluster

- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- Node Interfaces on Active SRX Series Chassis Clusters on page 846
- Node Interfaces on Active J Series Chassis Clusters on page 853
- Management Interface on an Active Chassis Cluster on page 855
- Fabric Interface on an Active Chassis Cluster on page 856
- Control Interface on an Active Chassis Cluster on page 856

## Understanding What Happens When Chassis Cluster Is Enabled

After wiring the two devices together as described in “Connecting SRX Series Hardware to Create a Chassis Cluster” on page 857 or “Connecting J Series Hardware to Create a Chassis Cluster” on page 863, you use CLI *operational mode* commands to enable chassis clustering by assigning a cluster ID and node ID on each chassis in the cluster. The cluster ID is the same on both nodes.

To do this, you connect to the console port on the primary device, give it a node ID, and identify the cluster it will belong to, and then reboot the system. You then connect the console port to the other device, give it a node ID, and assign it the same cluster ID you gave to the first node, and then reboot the system. In both instances, you can cause the system to boot automatically by including the **reboot** parameter in the CLI command line. (For further explanation of primary and secondary nodes, see “Understanding Chassis Cluster Redundancy Groups” on page 797.)



**CAUTION:** The factory default configuration for SRX100, SRX210, and SRX240 devices automatically enables Layer 2 Ethernet switching. Because Layer 2 Ethernet switching is not supported in chassis cluster mode, for these devices, if you use the factory default configuration, you must delete the Ethernet switching configuration before you enable chassis clustering. See “Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering” on page 860.



**CAUTION:** After fabric interfaces have been configured on a chassis cluster, removing the fabric configuration on either node will cause the redundancy group 0 (RG0) secondary node to move to a disabled state. (Resetting a device to the factory default configuration removes the fabric configuration and thereby causes the RG0 secondary node to move to a disabled state.) After the fabric configuration is committed, do not reset either device to the factory default configuration.

Figure 89 on page 851 shows how the FPC slots are numbered on two nodes in an SRX5000 line chassis cluster. Other figures show slot numbering on both nodes in other SRX Series chassis clusters. Figure 95 on page 855 shows how the PIM slots are numbered on two nodes in a J Series chassis cluster.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Node Interfaces on Active SRX Series Chassis Clusters on page 846
  - Node Interfaces on Active J Series Chassis Clusters on page 853
  - Management Interface on an Active Chassis Cluster on page 855
  - Fabric Interface on an Active Chassis Cluster on page 856
  - Control Interface on an Active Chassis Cluster on page 856
  - Disabling Chassis Cluster on page 877

## Node Interfaces on Active SRX Series Chassis Clusters

Normally, on SRX Series devices, the built-in interfaces are numbered as follows:

|                             |          |          |          |          |     |
|-----------------------------|----------|----------|----------|----------|-----|
| For Most SRX Series Devices | ge-0/0/0 | ge-0/0/1 | ge-0/0/2 | ge-0/0/3 | ... |
| For SRX210 Devices          | ge-0/0/0 | ge-0/0/1 | fe-0/0/2 | fe-0/0/3 | ... |

For SRX100  
Devices

fe-0/0/0

fe-0/0/1

fe-0/0/2

fe-0/0/3

...



**CAUTION:** Layer 2 switching must not be enabled on an SRX Series device when chassis clustering is enabled. If you have enabled Layer 2 switching, make sure you disable it before enabling chassis clustering.

The factory default configuration for SRX100, SRX210, and SRX240 devices automatically enables Layer 2 Ethernet switching. Because Layer 2 Ethernet switching is not supported in chassis cluster mode, for these devices, if you use the factory default configuration, you must delete the Ethernet switching configuration before you enable chassis clustering. See “Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering” on page 860.

For chassis clustering, all SRX Series devices have a built-in management interface name **fxp0**. For most SRX Series devices, the **fxp0** interface is a dedicated port. For SRX210 and SRX100 devices, after you enable chassis clustering and reboot the system, the built-in interface named **fe-0/0/6** is repurposed as the management interface and is automatically renamed **fxp0**.

For the SRX5000 line, control interfaces are configured on SPCs. For the SRX3000 line and the SRX650 and SRX240 devices, control interfaces are dedicated Gigabit Ethernet ports. For SRX210 and SRX100 devices, after you enable chassis clustering and reboot the system, the built-in interface named **fe-0/0/7** is repurposed as the control interface and is automatically renamed **fxp1**.

After the devices are connected as a cluster, the slot numbering on one device changes and thus the interface numbering will change. The slot number for each slot in both nodes is determined using the following formula:

**cluster slot number = (node ID \* maximum slots per node) + local slot number**

In chassis cluster mode, the interfaces on the secondary node are renumbered internally. For example, the management interface port on the front panel of each SRX210 device is still labeled **fe-0/0/6**, but internally, the node 1 port is referred to as **fe-2/0/6**.

Table 80 on page 848 shows the slot numbering, as well as the physical port and logical interface numbering, for both of the SRX Series devices that become node 0 and node 1 of the chassis cluster after the cluster is formed.

**Table 80: SRX Series Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming**

| Model | Chassis | Maximum Slots Per Node | Slot Numbering in a Cluster | Management Physical Port/Logical Interface | Control Physical Port/Logical Interface | Fabric Physical Port/Logical Interface |
|-------|---------|------------------------|-----------------------------|--------------------------------------------|-----------------------------------------|----------------------------------------|
| 5800  | Node 0  | 12 (FPC slots)         | 0 — 11                      | Dedicated Gigabit Ethernet port            | Control port on an SPC                  | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | em0                                     | fab0                                   |
|       | Node 1  |                        | 12 — 23                     | Dedicated Gigabit Ethernet port            | Control port on an SPC                  | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | em0                                     | fab1                                   |
| 5600  | Node 0  | 6 (FPC slots)          | 0 — 5                       | Dedicated Gigabit Ethernet port            | Control port on an SPC                  | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | em0                                     | fab0                                   |
|       | Node 1  |                        | 6 — 11                      | Dedicated Gigabit Ethernet port            | Control port on an SPC                  | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | em0                                     | fab1                                   |
| 3600  | Node 0  | 13 (CFM slots)         | 0 — 12                      | Dedicated Gigabit Ethernet port            | Dedicated Gigabit Ethernet port         | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | em0                                     | fab0                                   |
|       | Node 1  |                        | 13 — 25                     | Dedicated Gigabit Ethernet port            | Dedicated Gigabit Ethernet port         | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | em0                                     | fab1                                   |



**Table 80: SRX Series Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming** (*continued*)

| Model | Chassis | Maximum Slots Per Node | Slot Numbering in a Cluster | Management Physical Port/Logical Interface | Control Physical Port/Logical Interface | Fabric Physical Port/Logical Interface |
|-------|---------|------------------------|-----------------------------|--------------------------------------------|-----------------------------------------|----------------------------------------|
| 3400  | Node 0  | 8 (CFM slots)          | 0 — 7                       | Dedicated Gigabit Ethernet port            | Dedicated Gigabit Ethernet port         | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | em0                                     | fab0                                   |
|       | Node 1  |                        | 8 — 15                      | Dedicated Gigabit Ethernet port            | Dedicated Gigabit Ethernet port         | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | em0                                     | fab1                                   |
| 650   | Node 0  | 9 (PIM slots)          | 0 — 8                       | ge-0/0/0                                   | ge-0/0/1                                | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | fxp1                                    | fab0                                   |
|       | Node 1  |                        | 9 — 17                      | ge-9/0/0                                   | ge-9/0/1                                | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | fxp1                                    | fab1                                   |
| 240   | Node 0  | 5 (PIM slots)          | 0 — 4                       | ge-0/0/0                                   | ge-0/0/1                                | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | fxp1                                    | fab0                                   |
|       | Node 1  |                        | 5 — 9                       | ge-5/0/0                                   | ge-5/0/1                                | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | fxp1                                    | fab1                                   |
| 210   | Node 0  | 2 (PIM slots)          | 0 and 1                     | fe-0/0/6                                   | fe-0/0/7                                | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | fxp1                                    | fab0                                   |
|       | Node 1  |                        | 2 and 3                     | fe-2/0/6                                   | fe-2/0/7                                | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | fxp1                                    | fab1                                   |

**Table 80: SRX Series Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming (*continued*)**

| Model | Chassis | Maximum Slots Per Node | Slot Numbering in a Cluster | Management Physical Port/Logical Interface | Control Physical Port/Logical Interface | Fabric Physical Port/Logical Interface |
|-------|---------|------------------------|-----------------------------|--------------------------------------------|-----------------------------------------|----------------------------------------|
| 100   | Node 0  | 1(PIM slot)            | 0                           | fe-0/0/6                                   | fe-0/0/7                                | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | fxp1                                    | fab0                                   |
|       | Node 1  |                        | 1                           | fe-1/0/6                                   | fe-1/0/7                                | Any Ethernet port                      |
|       |         |                        |                             | fxp0                                       | fxp1                                    | fab1                                   |

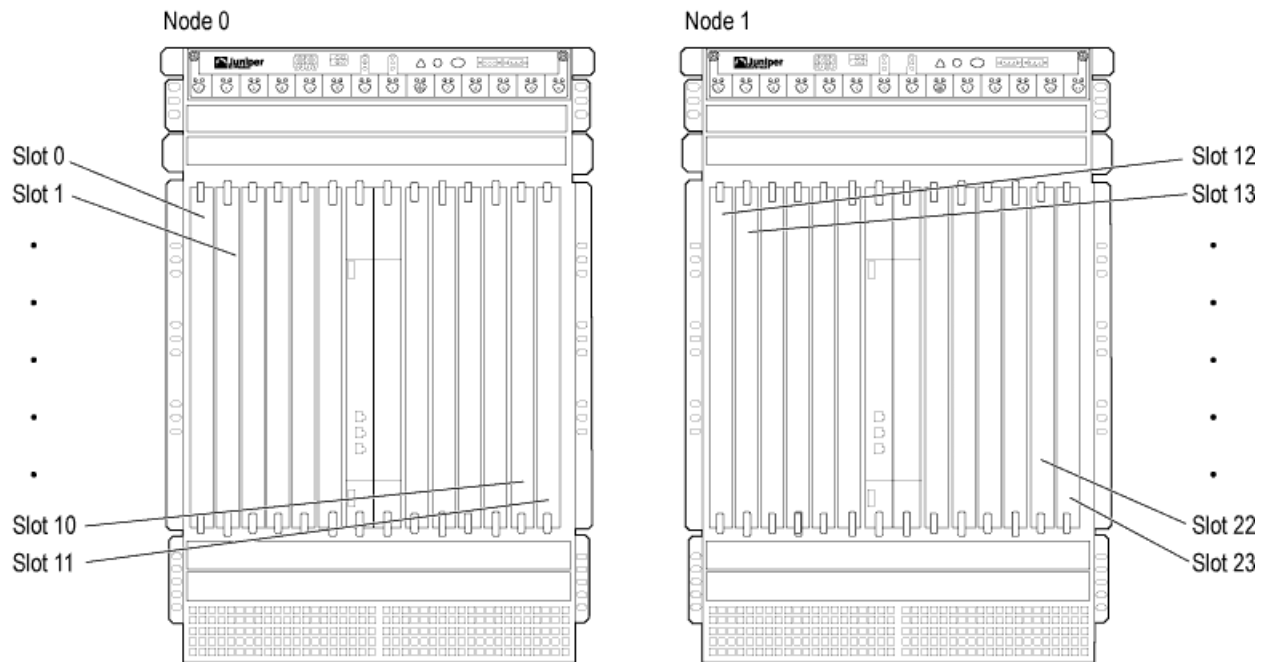
Information about chassis cluster slot numbering is also provided in Figure 89 on page 851, Figure 90 on page 852, Figure 91 on page 852, Figure 92 on page 852, Figure 93 on page 853, and Figure 94 on page 853



**NOTE:** See the appropriate *Services Gateway Hardware Guide* for details about SRX Series devices. The *Junos OS Interfaces Configuration Guide for Security Devices* provides a full discussion of the interface naming convention.

After you enable chassis clustering, the two chassis joined together cease to exist as individuals and now represent a single system. As a single system, the cluster now has twice as many slots. (See Figure 89 on page 851, Figure 90 on page 852, Figure 91 on page 852, Figure 92 on page 852, Figure 93 on page 853, and Figure 94 on page 853.)

Figure 89: FPC Slot Numbering in an SRX Series Chassis Cluster (SRX5800 Devices)



NOTE: SRX5600 and SRX5800 devices have Flex I/O Cards (Flex IOCs) that have two slots to accept the following port modules:

- SRX-IOC-4XGE-XFP 4-Port XFP
- SRX-IOC-16GE-TX 16-Port RJ-45
- SRX-IOC-16GE-SFP 16-Port SFP

You can use these port modules to add from 4 to 16 Ethernet ports to your SRX Series device. Port numbering for these modules is

*slot/port module/port*

where *slot* is the number of the slot in the device in which the Flex IOC is installed; *port module* is 0 for the upper slot in the Flex IOC or 1 for the lower slot when the card is vertical, as in an SRX5800 device; and *port* is the number of the port on the port module. When the card is horizontal, as in an SRX5600 device, *port module* is 0 for the left-hand slot or 1 for the right-hand slot.

See the *Services Gateway Hardware Guide* for your specific SRX Series model.

Figure 90: Slot Numbering in an SRX Series Chassis Cluster (SRX3400 Devices)

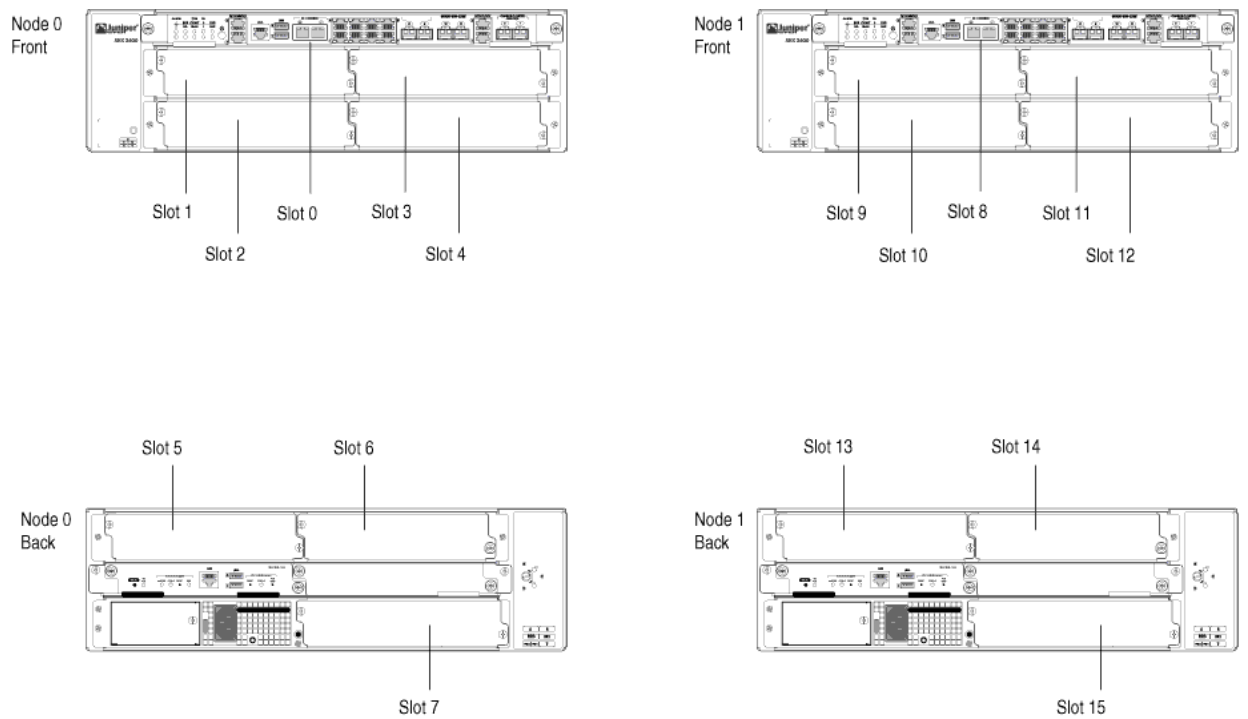


Figure 91: Slot Numbering in an SRX Series Chassis Cluster (SRX650 Devices)

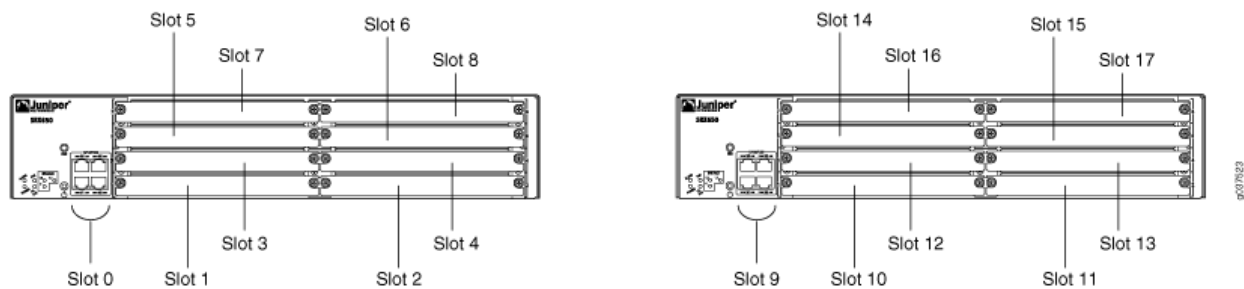


Figure 92: Slot Numbering in an SRX Series Chassis Cluster (SRX240 Devices)

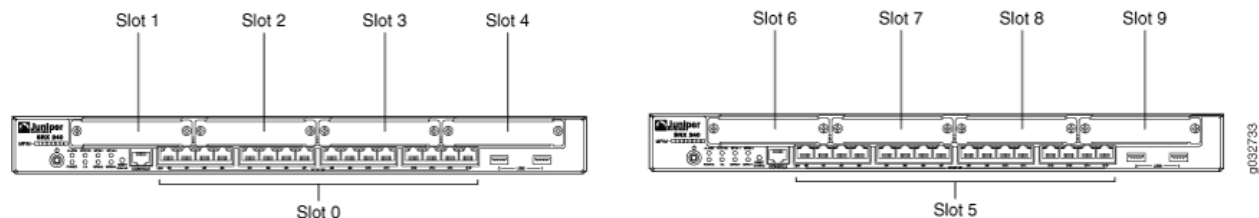


Figure 93: Slot Numbering in an SRX Series Chassis Cluster (SRX210 Devices)

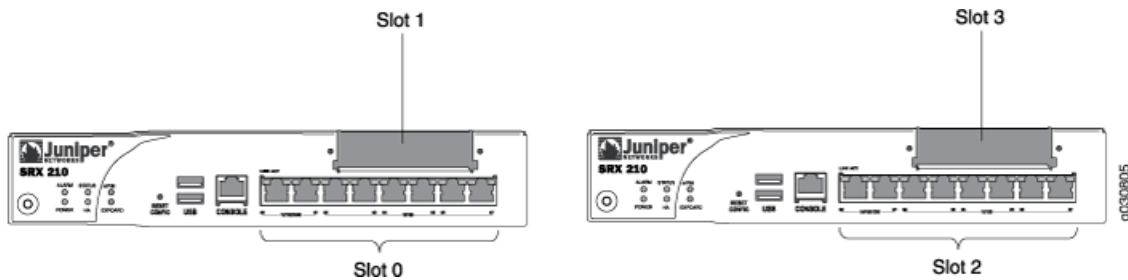
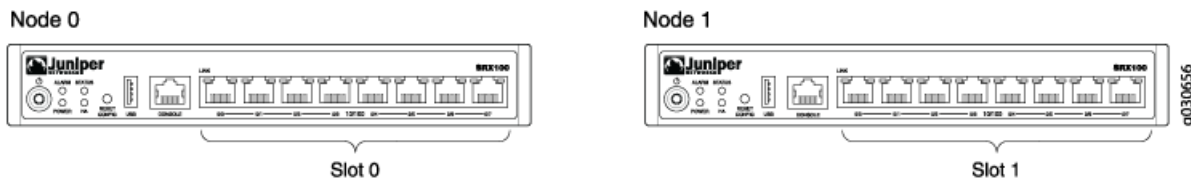


Figure 94: Slot Numbering in an SRX Series Chassis Cluster (SRX100 Devices)



- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding What Happens When Chassis Cluster Is Enabled on page 845
  - Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering on page 860
  - Node Interfaces on Active J Series Chassis Clusters on page 853
  - Management Interface on an Active Chassis Cluster on page 855
  - Fabric Interface on an Active Chassis Cluster on page 856
  - Control Interface on an Active Chassis Cluster on page 856

## Node Interfaces on Active J Series Chassis Clusters

Normally, on J Series devices, the built-in interfaces are numbered as follows:

ge-0/0/0      ge-0/0/1      ge-0/0/2      ge-0/0/3      ...



**CAUTION:** Layer 2 switching must not be enabled on J Series device when chassis clustering is enabled. If you have enabled Layer 2 switching, make sure you disable it before enabling chassis clustering.

After you enable chassis clustering and reboot the system, the built-in interface named ge-0/0/2 is repurposed as the management interface and is automatically renamed **fxp0**. Likewise, the built-in interface named ge-0/0/3 is repurposed as the control interface and is automatically renamed **fxp1**.

After the devices are connected as a cluster, the slot numbering and thus the interface numbering will change for one device. The cluster determines the slot number for each slot in both nodes using the following formula:

$$\text{cluster slot number} = (\text{node ID} * \text{maximum slots per node}) + \text{local slot number}$$

In chassis cluster mode, the interfaces on the secondary node are renumbered internally. For example, the management interface port on the front panel of each J2320 device is still labeled ge-0/0/2, but internally, the node 1 port is referred to as ge-4/0/2.

Table 81 on page 854 shows the slot numbering, as well as the port and interface numbering, for both of the J Series devices that become node 0 and node 1 of the cluster after the cluster is formed.

**Table 81: J Series Chassis Cluster Slot Numbering, and Physical Port and Logical Interface Naming**

| Model           | Chassis | Maximum Slots Per Node                            | Slot Numbering in a Cluster | Management Physical Port/Logical Interface | Control Physical Port/Logical Interface | Fabric Physical Port/Logical Interface |
|-----------------|---------|---------------------------------------------------|-----------------------------|--------------------------------------------|-----------------------------------------|----------------------------------------|
| J2320           | Node 0  | 4 (PIM slots);<br><i>includes one preset slot</i> | 0 — 3                       | ge-0/0/2                                   | ge-0/0/3                                | Any Gigabit Ethernet port              |
|                 |         |                                                   |                             | fxp0                                       | fxp1                                    | fab0                                   |
|                 | Node 1  |                                                   | 4 — 7                       | ge-4/0/2                                   | ge-4/0/3                                | Any Gigabit Ethernet port              |
|                 |         |                                                   |                             | fxp0                                       | fxp1                                    | fab1                                   |
| J2350           | Node 0  | 6 (PIM slots);<br><i>includes one preset slot</i> | 0 — 5                       | ge-0/0/2                                   | ge-0/0/3                                | Any Gigabit Ethernet port              |
|                 |         |                                                   |                             | fxp0                                       | fxp1                                    | fab0                                   |
|                 | Node 1  |                                                   | 6 — 11                      | ge-6/0/2                                   | ge-6/0/3                                | Any Gigabit Ethernet port              |
|                 |         |                                                   |                             | fxp0                                       | fxp1                                    | fab1                                   |
| J4350 and J6350 | Node 0  | 7 (PIM slots);<br><i>includes one preset slot</i> | 0 — 6                       | ge-0/0/2                                   | ge-0/0/3                                | Any Gigabit Ethernet port              |
|                 |         |                                                   |                             | fxp0                                       | fxp1                                    | fab0                                   |
|                 | Node 1  |                                                   | 7 — 13                      | ge-7/0/2                                   | ge-7/0/3                                | Any Gigabit Ethernet port              |
|                 |         |                                                   |                             | fxp0                                       | fxp1                                    | fab1                                   |

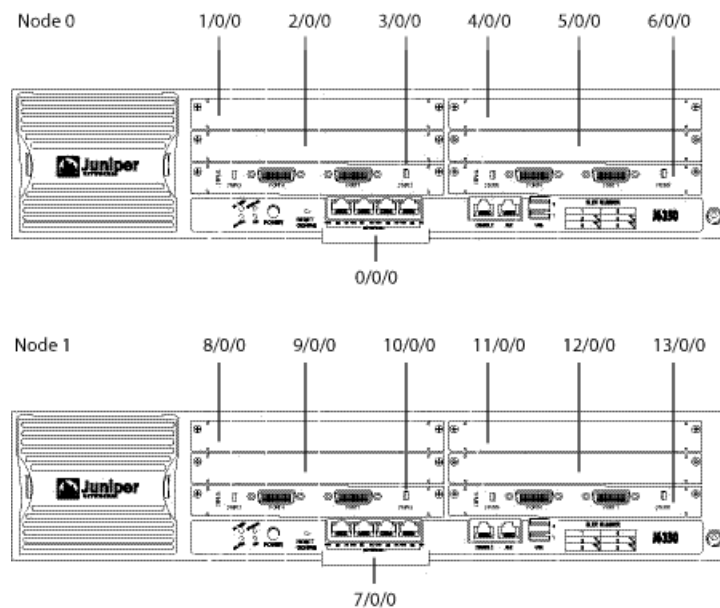
Information about chassis cluster slot numbering is also provided in Figure 95 on page 855.



NOTE: See the *J Series Services Routers Hardware Guide* for details about J Series devices. The *Junos OS Interfaces Configuration Guide for Security Devices* provides a full discussion of the interface naming convention.

After you enable chassis clustering, the two chassis joined together cease to exist as individuals and now represent a single system. As a single system, the cluster now has twice as many PIM slots. (See Figure 95 on page 855.)

**Figure 95: PIM Slot Numbering in a J Series Chassis Cluster (J6350 Devices)**



- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding What Happens When Chassis Cluster Is Enabled on page 845
  - Node Interfaces on Active SRX Series Chassis Clusters on page 846
  - Management Interface on an Active Chassis Cluster on page 855
  - Fabric Interface on an Active Chassis Cluster on page 856
  - Control Interface on an Active Chassis Cluster on page 856

## Management Interface on an Active Chassis Cluster

The **fxp0** interfaces function like standard management interfaces on SRX Series and J Series devices and allow network access to each node in the cluster. You must, however, first connect to each node through the console port and assign a unique IP address to each **fxp0** interface.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding What Happens When Chassis Cluster Is Enabled on page 845

- [Node Interfaces on Active SRX Series Chassis Clusters on page 846](#)
- [Node Interfaces on Active J Series Chassis Clusters on page 853](#)
- [Fabric Interface on an Active Chassis Cluster on page 856](#)
- [Control Interface on an Active Chassis Cluster on page 856](#)

## Fabric Interface on an Active Chassis Cluster

The fabric is the data link between the nodes and is used to forward traffic between the chassis. Traffic arriving on a node that needs to be processed on the other is forwarded over the fabric data link. Similarly, traffic processed on a node that needs to exit through an interface on the other node is forwarded over the fabric. The fabric also provides for synchronization of session state objects created by operations such as authentication, Network Address Translation (NAT), Application Layer Gateways (ALGs), and IP Security (IPsec) sessions. For SRX Series chassis clusters, the fabric link can be any pair of Ethernet interfaces spanning the cluster; for J Series chassis clusters, the fabric link can be any pair of Gigabit Ethernet interfaces.

- Related Topics**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding What Happens When Chassis Cluster Is Enabled on page 845](#)
  - [Node Interfaces on Active SRX Series Chassis Clusters on page 846](#)
  - [Node Interfaces on Active J Series Chassis Clusters on page 853](#)
  - [Management Interface on an Active Chassis Cluster on page 855](#)
  - [Control Interface on an Active Chassis Cluster on page 856](#)

## Control Interface on an Active Chassis Cluster

The control interfaces provide the control link between the two nodes in the cluster and are used for routing updates and for control plane signal traffic, such as heartbeat and threshold information that triggers node failover. The control link is also used to synchronize the configuration between the nodes. When you submit configuration statements to the cluster, the configuration is automatically synchronized over the control link.

- Related Topics**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Understanding What Happens When Chassis Cluster Is Enabled on page 845](#)
  - [Node Interfaces on Active SRX Series Chassis Clusters on page 846](#)
  - [Node Interfaces on Active J Series Chassis Clusters on page 853](#)
  - [Management Interface on an Active Chassis Cluster on page 855](#)
  - [Fabric Interface on an Active Chassis Cluster on page 856](#)



## Building a Chassis Cluster

- Connecting SRX Series Hardware to Create a Chassis Cluster on page 857
- Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering on page 860
- SRX Series Chassis Cluster Configuration Overview on page 861
- Connecting J Series Hardware to Create a Chassis Cluster on page 863
- J Series Chassis Cluster Configuration Overview on page 864
- Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
- Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
- Verifying a Chassis Cluster Configuration on page 869
- Verifying Chassis Cluster Statistics on page 870
- Clearing Chassis Cluster Statistics on page 871
- Verifying Chassis Cluster Failover Status on page 872
- Clearing Chassis Cluster Failover Status on page 873

## Connecting SRX Series Hardware to Create a Chassis Cluster

An SRX Series chassis cluster is created by physically connecting two identical cluster-supported SRX Series devices together using a pair of the same type of Ethernet connections. The connection is made for both a control link and a fabric (data) link between the two devices.



**NOTE:** You can connect two control links (SRX5000 and SRX3000 lines only) and two fabric links between the two devices in the cluster to reduce the chance of control link and fabric link failure. See “Understanding Chassis Cluster Dual Control Links” on page 830 and “Understanding Chassis Cluster Dual Fabric Links” on page 841.

Control links in a chassis cluster are made using specific ports.

You must use the following ports to form the control link on the branch SRX Series devices:

- For SRX100 devices, connect the fe-0/0/7 on node 0 to the fe-1/0/7 on node 1.
- For SRX210 devices, connect the fe-0/0/7 on node 0 to the fe-2/0/7 on node 1.
- For SRX240 devices, connect the ge-0/0/1 on node 0 to the ge-5/0/1 on node 1.
- For SRX650 devices, connect the ge-0/0/1 on node 0 to the ge-9/0/1 on node 1.

For a device from the SRX3000 line, the connection that serves as the control link must be between the built-in control ports on each device.

SRX5000 line devices do not have built-in ports, so the control link for these gateways must be the control ports on their Services Processing Cards (SPCs) with a slot numbering offset of 6 for SRX5600 devices and 12 for SRX5800 devices.

When you connect a single control link on SRX3000 or SRX5000 line devices, the control link ports are a one-to-one mapping with the Routing Engine slot. If your Routing Engine is in slot 0, you must use control port 0 to link the Routing Engines.

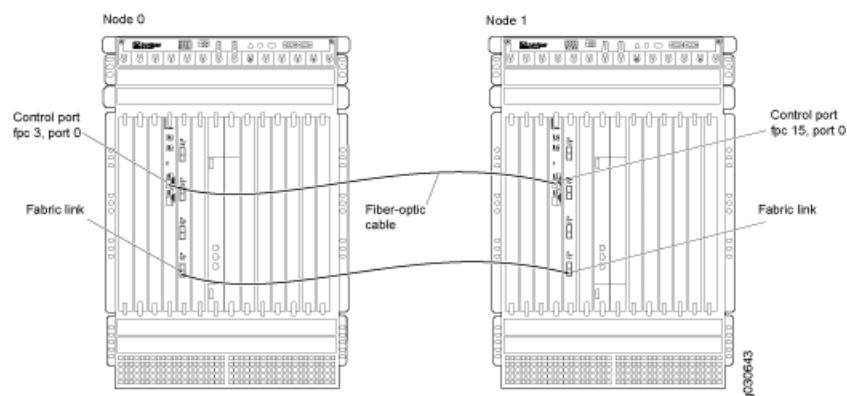


**NOTE:** For dual control links on SRX3000 line devices, the Routing Engine must be in slot 0 and the SRX Clustering Module (SCM) in slot 1. The opposite configuration (SCM in slot 0 and Routing Engine in slot 1) is not supported.

The fabric link connection for the SRX100 must be a pair of Fast Ethernet interfaces and for the SRX210 must be a pair of either Fast Ethernet or Gigabit Ethernet interfaces. The fabric link connection must be any pair of either Gigabit Ethernet or 10-Gigabit Ethernet interfaces on all other SRX Series devices.

Figure 96 on page 858, Figure 97 on page 858, Figure 98 on page 859, Figure 99 on page 859, Figure 100 on page 859, and Figure 101 on page 859 show pairs of SRX Series devices with the fabric links and control links connected.

**Figure 96: Connecting SRX Series Devices in a Cluster (SRX5800 Devices)**



**Figure 97: Connecting SRX Series Devices in a Cluster (SRX3400 Devices)**

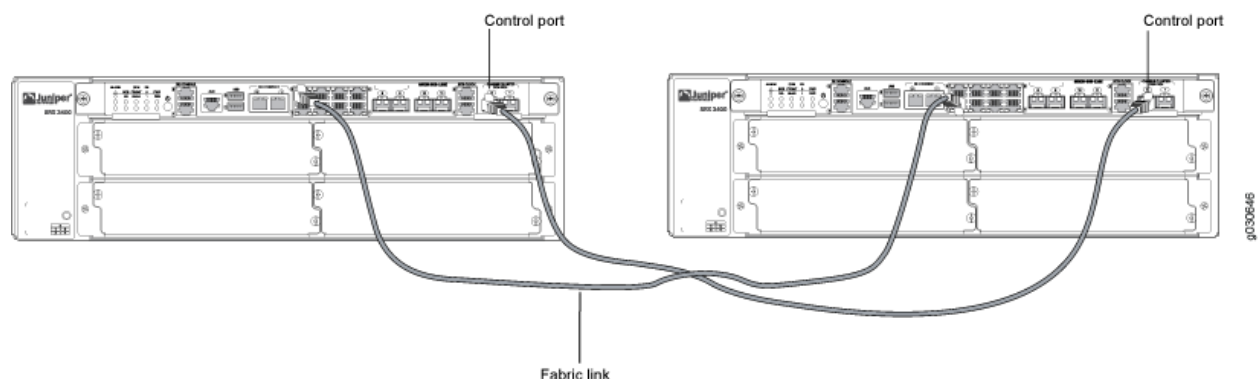


Figure 98: Connecting SRX Series Devices in a Cluster (SRX650 Devices)

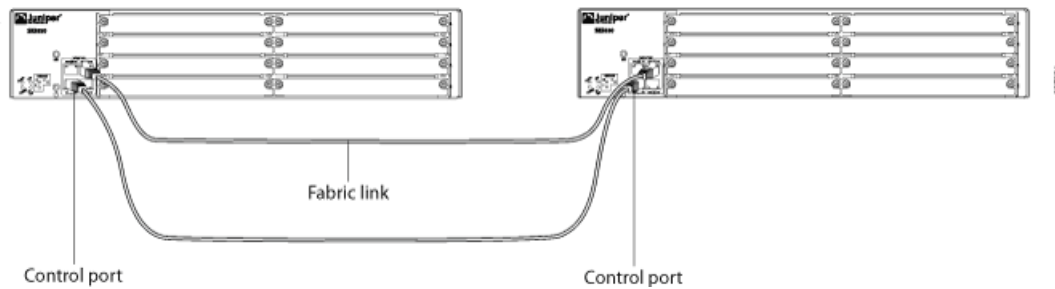


Figure 99: Connecting SRX Series Devices in a Cluster (SRX240 Devices)

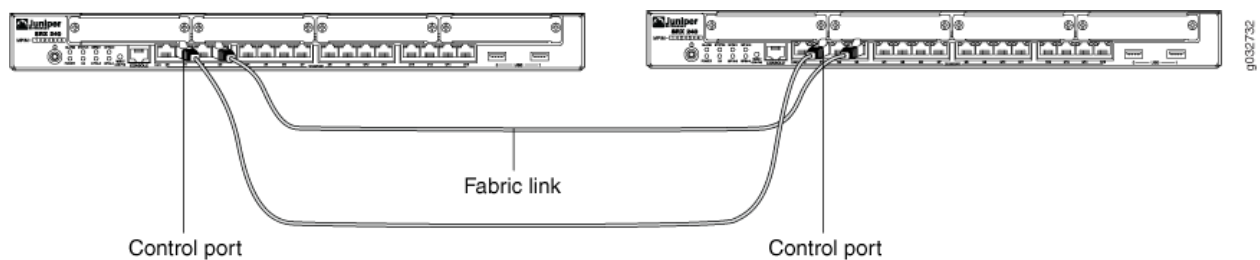


Figure 100: Connecting SRX Series Devices in a Cluster (SRX210 Devices)

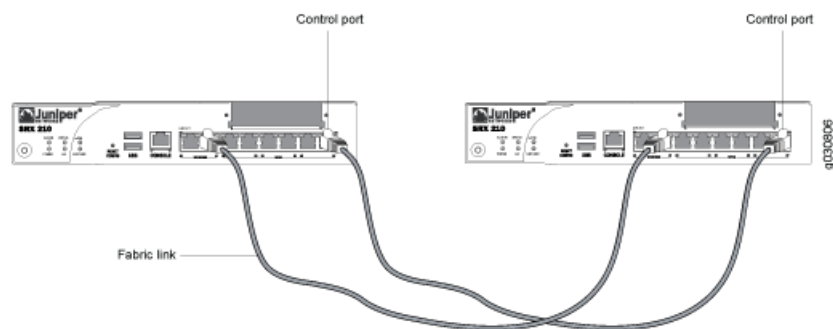
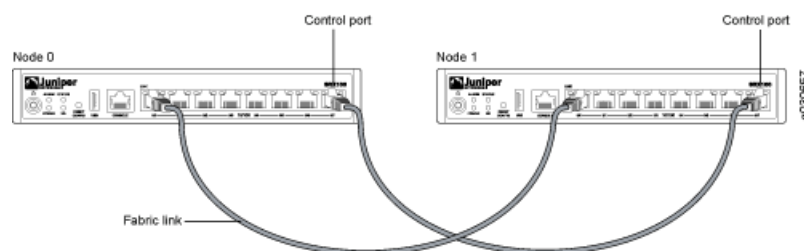


Figure 101: Connecting SRX Series Devices in a Cluster (SRX100 Devices)



- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding What Happens When Chassis Cluster Is Enabled on page 845
  - SRX Series Chassis Cluster Configuration Overview on page 861
  - Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering on page 860

- Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
- Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869

## Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering

The factory default configuration for SRX100, SRX210, and SRX240 devices automatically enables Layer 2 Ethernet switching. Because Layer 2 Ethernet switching is not supported in chassis cluster mode, for these devices, if you use the factory default configuration, you must delete the Ethernet switching configuration before you enable chassis clustering.



**CAUTION:** Enabling chassis clustering while Ethernet switching is enabled is not a supported configuration. Doing so might result in undesirable behavior from the devices, leading to possible network instability.

Specifically, the factory default configuration includes virtual LAN (VLAN) configuration, and the chassis cluster control link is Ethernet switching enabled. To use the control link, the Ethernet switching family must be disabled on the interface.

The following procedure shows how to configure chassis clustering on SRX100, SRX210, and SRX240 devices from the factory default configuration. Follow the procedure before starting the chassis cluster configuration for each of the two devices to be clustered.

We recommend that you do these steps through a console port connection, but if you do not, you will have to connect through the console after you complete the steps. For information about how to connect through the console, see the “Connecting and Configuring the Device” section in the appropriate *SRX Series Services Gateway Getting Started Guide*.

1. Enter configuration mode.
2. Enter the following commands:

```
user@host# set system root-authentication plain-text-password
```

This setting is required if a root user password was not set.

```
user@host# delete vlans
user@host# delete interfaces vlan
user@host# delete interfaces interface-range interfaces-trust
user@host# delete security zones security-zone trust interfaces
user@host# commit
```

Note that once you commit this configuration, the management interfaces will be lost and need to be recreated.

Likewise, if you are not using the factory default configuration on these devices (SRX100, SRX210, and SRX240), but you enable Ethernet switching on the interfaces, be sure to disable Ethernet switching before you enable chassis clustering.



**NOTE:** The default configuration for other SRX Series devices and all J Series devices does not automatically enable Ethernet switching. However, if you have enabled Ethernet switching, be sure to disable it before enabling clustering on these devices too.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- Node Interfaces on Active SRX Series Chassis Clusters on page 846
- Node Interfaces on Active J Series Chassis Clusters on page 853
- Management Interface on an Active Chassis Cluster on page 855
- Control Interface on an Active Chassis Cluster on page 856

## SRX Series Chassis Cluster Configuration Overview

This section provides an overview of the basic steps to create an SRX Series chassis cluster.

For the basic steps to set up a J Series chassis cluster, see “J Series Chassis Cluster Configuration Overview” on page 864.

Before you begin, connect the SRX Series devices using the instructions in “Connecting SRX Series Hardware to Create a Chassis Cluster” on page 857



**NOTE:** For SRX5000 line chassis clusters, the placement and type of SPCs must match in the two devices. For SRX3000 line chassis clusters, the placement and type of SPCs, IOC, and NPC must match in the two devices. For SRX650, SRX240, SRX210, and SRX100 chassis clusters, the placement and type of GPIMs, XGPIMs, XPIMs, and Mini-PIMs (as applicable) must match in the two devices.

To create an SRX Series chassis cluster:

1. Physically connect a pair of the same kind of supported SRX Series devices together:
  - a. Create the fabric link between two nodes in a cluster by connecting any pair of Ethernet interfaces. For most SRX Series devices, the only requirement is that both interfaces be Gigabit Ethernet interfaces (or 10-Gigabit Ethernet interfaces). For SRX210 devices, both interfaces must be of a similar type (that is, a pair of Fast Ethernet interfaces or a pair of Gigabit Ethernet interfaces). For SRX100 devices, connect a pair of Fast Ethernet interfaces. Figure 96 on page 858 shows nodes connected using built-in I/O ports for the fabric link.

When using dual fabric link functionality, connect the two pairs of Ethernet interfaces that you will use on each device. See “Understanding Chassis Cluster Dual Fabric Links” on page 841.

- b. Connect the control ports that you will use on each device (for example, **fpc3** and **fpc15**, as shown in Figure 96 on page 858). For SRX3600, SRX3400, SRX650, and SRX240 devices, the control ports are dedicated Gigabit Ethernet ports. For SRX210 and SRX100 devices, the control port is the highest numbered port (**fe-0/0/7**).

When using dual control link functionality (SRX5000 and SRX3000 lines only), connect the two pairs of control ports that you will use on each device (for example, **fpc3** and **fpc15** for the first pair and **fpc6** and **fpc18** for the second, as shown in Figure 87 on page 832). See “Connecting Dual Control Links for SRX Series Devices in a Chassis Cluster” on page 832.

For SRX5600 and SRX5800 devices, control ports should be on corresponding slots in the two devices, with the following slot numbering offsets:

| Device  | Offset                                          |
|---------|-------------------------------------------------|
| SRX5800 | 12 (for example, <b>fpc3</b> and <b>fpc15</b> ) |
| SRX5600 | 6 (for example, <b>fpc3</b> and <b>fpc9</b> )   |

2. Connect the first device to be initialized in the cluster to the console port. This is the node that forms the cluster.

For connection instructions, see the appropriate *Services Gateway Getting Started Guide*.

3. Configure the control ports (SRX5000 line only). See “Example: Configuring Chassis Cluster Control Ports (CLI)” on page 830.
4. Use CLI operational mode commands to enable clustering:
  - a. Identify the cluster by giving it the cluster ID.
  - b. Identify the node by giving it its own node ID and then reboot the system.

See “Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI)” on page 866.

5. Connect to the console port on the other device and use CLI operational mode commands to enable clustering:
  - a. Identify the cluster that the device is joining by setting the same cluster ID you set on the first node.
  - b. Identify the node by giving it its own node ID and then reboot the system.
6. Configure the management interfaces on the cluster. See “Example: Configuring the Chassis Cluster Management Interface (CLI)” on page 867.
7. Configure the cluster with the CLI. See:

- a. Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
  - b. Example: Configuring the Chassis Cluster Fabric (CLI) on page 842
  - c. Example: Configuring Chassis Cluster Redundancy Groups (CLI) on page 803
  - d. Example: Configuring Chassis Cluster Redundant Ethernet Interfaces (CLI) on page 819
  - e. Example: Configuring Chassis Cluster Interface Monitoring (CLI) on page 805
8. Initiate manual failover. See “Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 814.
  9. Configure conditional route advertisement over redundant Ethernet interfaces. See “Understanding Conditional Route Advertising in a Chassis Cluster” on page 825.
  10. Verify the configuration. See “Verifying a Chassis Cluster Configuration” on page 869.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- Connecting SRX Series Hardware to Create a Chassis Cluster on page 857
- Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
- Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
- Verifying a Chassis Cluster Configuration on page 869

### Connecting J Series Hardware to Create a Chassis Cluster

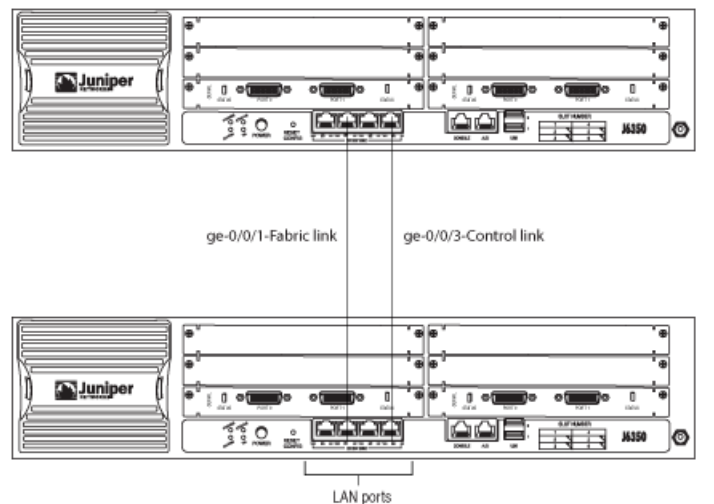
To create a J Series chassis cluster, you must physically connect a pair of the same kind of supported J Series devices back-to-back over a pair of Gigabit Ethernet connections. The connection that serves as the control link must be the built-in interface **ge-0/0/3**. The fabric link connection can be a combination of any pair of Gigabit Ethernet interfaces on the devices.



**NOTE:** You can connect two fabric links between the two devices in the cluster to reduce the chance of fabric link failure. See “Understanding Chassis Cluster Dual Fabric Links” on page 841.

Figure 102 on page 864 shows two J Series devices connected using the built-in interfaces for both the fabric and control links.

Figure 102: Connecting J Series Devices in a Cluster (J6350 Devices)



NOTE: When chassis clustering is enabled on a J Series router, two interface ports are used to link the two devices: the ge-0/0/3 interface (fxp1 port) is used for the control interface and one port is used for the fabric link (using either one of the built-in interfaces (ge-0/0/0 or ge-0/0/1) or one of the ports of a uPIM). Also, the ge-0/0/2 interface (fxp0 port) is used for the management link. This means that three of the four onboard Gigabit Ethernet ports are in use; if additional ports are required for transit traffic, then a PIM or uPIM is required.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- J Series Chassis Cluster Configuration Overview on page 864
- Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
- Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
- Verifying a Chassis Cluster Configuration on page 869

### J Series Chassis Cluster Configuration Overview

This section provides an overview of the basic steps to create a J Series chassis cluster.

For the basic steps to set up an SRX Series chassis cluster, see “SRX Series Chassis Cluster Configuration Overview” on page 861.

Before you begin, connect the J Series devices using the instructions in “Connecting J Series Hardware to Create a Chassis Cluster” on page 863





**NOTE:** For J Series chassis clusters, the two nodes in a cluster must be identical models, but can have any combination of PIMs installed.

To create a J Series chassis cluster:

1. Physically connect a pair of the same kind of supported J Series devices together:
  - a. Create the fabric link between two nodes in a cluster by connecting any pair of Gigabit Ethernet interfaces, either the built-in interfaces or interfaces on the PIMs. The only requirement is that both interfaces be Gigabit Ethernet interfaces. Figure 102 on page 864 shows nodes connected using the built-in **ge-0/0/1** interface for the fabric link.

When using dual fabric link functionality, connect the two pairs of Gigabit Ethernet interfaces that you will use on each device. For more information, see “Understanding Chassis Cluster Dual Fabric Links” on page 841.

- b. Connect the **ge-0/0/3** interfaces together to create the control link.
2. Connect the first device to be initialized in the cluster to the console port. This is the node that forms the cluster.

For connection instructions, see the *J Series Services Routers Hardware Guide*.

3. Use CLI operational mode commands to enable clustering:
  - a. Identify the cluster by giving it a cluster ID.
  - b. Identify the node by giving it its own node ID and then reboot the system.

See “Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI)” on page 866.

4. Connect to the console port on the other device and use CLI operational mode commands to enable clustering:
  - a. Identify the cluster that the device is joining by setting the same cluster ID you set on the first node.
  - b. Identify the node by giving it its own node ID and then reboot the system.
5. Configure the management interfaces on the cluster. See “Example: Configuring the Chassis Cluster Management Interface (CLI)” on page 867.
6. Configure the cluster with the CLI. See:
  - a. Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
  - b. Example: Configuring the Chassis Cluster Fabric (CLI) on page 842
  - c. Example: Configuring Chassis Cluster Redundancy Groups (CLI) on page 803
  - d. Example: Configuring Chassis Cluster Redundant Ethernet Interfaces (CLI) on page 819
  - e. Example: Configuring Chassis Cluster Interface Monitoring (CLI) on page 805

7. Initiate manual failover. See “Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 814.
8. Configure conditional route advertisement over redundant Ethernet interfaces. See “Understanding Conditional Route Advertising in a Chassis Cluster” on page 825.
9. Verify the configuration. See “Verifying a Chassis Cluster Configuration” on page 869.

**Related Topics**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- Connecting J Series Hardware to Create a Chassis Cluster on page 863
- Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
- Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
- Verifying a Chassis Cluster Configuration on page 869

### Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI)

After connecting the two devices together, you configure a cluster ID and a node ID. A cluster ID identifies the cluster that the two nodes belong to. A node ID identifies a unique node within a cluster.

Before you begin, disable switching on the SRX100, SRX210, and SRX240 devices using the instructions in “Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering” on page 860.

The system uses node IDs and cluster IDs to apply the correct configuration for each node when you use the **apply-group** command described in “Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI)” on page 869. The node ID and cluster ID statements are written to the EPROM, and when the system is rebooted, they take effect.

You can deploy up to 15 clusters in a Layer 2 domain. Each cluster is defined by a **cluster-id** value within the range of 1 through 15. A device can belong to only one cluster at any given time. Nodes in a cluster are numbered 0 and 1.

To set the node IDs and cluster IDs, connect to each device through the console port, enter the following operational commands, and then reboot the system.

- Enter the cluster ID and node ID information for the first node. If you want redundancy groups to be primary on this node when priority settings for both nodes are the same, make it node 0.

```
user@host> set chassis cluster cluster-id 1 node 0
warning: A reboot is required for chassis cluster to be enabled
```

- Enter the cluster ID and node ID for the other node. If you want redundancy groups to be secondary on this node when priority settings for both nodes are the same, make it node 1.

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
Successfully enabled chassis cluster. Going to reboot now.
```

After you set the cluster ID and node ID for each node and the system reboots, the built-in interfaces are automatically renamed (see Table 81 on page 854). Use the **show chassis cluster status** operational command to view node status.

```
{primary:node1}
user@host# show chassis cluster status
Cluster ID: 3
Node name Priority Status Preempt Manual failover

Redundancy group: 0 , Failover count: 1
 node0 254 secondary no no
 node1 2 primary no no

Redundancy group: 1 , Failover count: 1
 node0 101 Secondary no no
 node1 99 primary no no
```

When you complete the chassis cluster basic configuration, any subsequent configuration changes you make are automatically synchronized on both nodes.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- SRX Series Chassis Cluster Configuration Overview on page 861
- Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering on page 860
- J Series Chassis Cluster Configuration Overview on page 864
- Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869

### Example: Configuring the Chassis Cluster Management Interface (CLI)

You must assign a unique IP address to each node in the cluster to provide network management access. This configuration is not replicated across the two nodes.

Before you begin, set the chassis cluster node ID and cluster ID using the instructions in “Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI)” on page 866

In most SRX Series chassis clusters, the **fxp0** interface is a dedicated port. For SRX210, SRX100, and all J Series chassis clusters, the **fxp0** interface is repurposed from a built-in interface.

In a J Series chassis cluster, you configure management access to the cluster by defining a unique hostname for each node and assigning a unique IP address to the **fxp0** interface

on each node. The **fxp0** interface is created when the system reboots the devices after you designate one node as the primary device and the other as the secondary device.



**NOTE:** If you try to access the nodes in a cluster over the network before you configure the **fxp0** interface, you will lose access to the cluster.

From the console port connection to the device you want to designate as the primary node, in configuration mode enter the following commands to name the node **node0-router** and assign IP address **10.1.1.1/24** to it:

```
user@host# set groups node0 system host-name node0-router
user@host# set groups node0 interfaces fxp0 unit 0 family inet address 10.1.1.1/24
```

For IPv6:

```
user@host# set groups node0 system host-name node0-router
user@host# set groups node0 interfaces fxp0 unit 0 family inet6 address
2010:2010:201::2/64
```

From the console port connection of the device you want to designate the secondary node, in configuration mode enter the following commands to name the node **node1-router** and assign IP address **10.1.1.2/24** to it:

```
user@host# set groups node1 system host-name node1-router
user@host# set groups node1 interfaces fxp0 unit 0 family inet address 10.1.1.2/24
```

For IPv6:

```
user@host# set groups node1 system host-name node1-router
user@host# set groups node1 interfaces fxp0 unit 0 family inet6 address
2010:2010:201::3/64
```

Enter the following command in configuration mode to apply these unique configurations to the appropriate node. (If you are migrating from a device to a cluster, delete the hostname from the configuration and then use the **apply-groups** command.)

This configuration is not replicated across the two nodes.

```
user@host# set apply-groups "${node}"
```

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- SRX Series Chassis Cluster Configuration Overview on page 861
- J Series Chassis Cluster Configuration Overview on page 864
- Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
- Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
- Verifying a Chassis Cluster Configuration on page 869

## Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI)

For the chassis cluster configuration, you specify the number of redundant Ethernet interfaces that the cluster contains and the information used to monitor the “health” of the cluster.

Before you begin, set the chassis cluster node ID and cluster ID using the instructions in “Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI)” on page 866

You must configure the redundant Ethernet interfaces count for the cluster in order for the redundant Ethernet interfaces that you configure to be recognized.

Use the following command in configuration mode to define the number of redundant Ethernet interfaces for the cluster:

```
{primary:node1}
user@host# set chassis cluster reth-count 3
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding What Happens When Chassis Cluster Is Enabled on page 845
  - SRX Series Chassis Cluster Configuration Overview on page 861
  - J Series Chassis Cluster Configuration Overview on page 864
  - Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
  - Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
  - Verifying a Chassis Cluster Configuration on page 869

## Verifying a Chassis Cluster Configuration

**Purpose** Display chassis cluster verification options.

**Action** From the CLI, enter the **show chassis cluster ?** command:

```
{primary:node1}
user@host> show chassis cluster ?
Possible completions:
 interfaces Display chassis-cluster interfaces
 statistics Display chassis-cluster traffic statistics
 status Display chassis-cluster status
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
  - Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
  - Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
  - Verifying Chassis Cluster Statistics on page 870
  - Clearing Chassis Cluster Statistics on page 871

- Verifying Chassis Cluster Failover Status on page 872

## Verifying Chassis Cluster Statistics

**Purpose** Display information about chassis cluster services and interfaces.

**Action** From the CLI, enter the **show chassis cluster statistics** command:

```
{primary:node1}
user@host> show chassis cluster statistics

Control link statistics:
 Control link 0:
 Heartbeat packets sent: 798
 Heartbeat packets received: 784
Fabric link statistics:
 Probes sent: 793
 Probes received: 0
Services Synchronized:
 Service name RT0s sent RT0s received
 Translation context 0 0
 Incoming NAT 0 0
 Resource manager 0 0
 Session create 0 0
 Session close 0 0
 Session change 0 0
 Gate create 0 0
 Session ageout refresh requests 0 0
 Session ageout refresh replies 0 0
 IPSec VPN 0 0
 Firewall user authentication 0 0
 MGCP ALG 0 0
 H323 ALG 0 0
 SIP ALG 0 0
 SCCP ALG 0 0
 PPTP ALG 0 0
 RTSP ALG 0 0

{primary:node1}
user@host> show chassis cluster statistics

Control link statistics:
 Control link 0:
 Heartbeat packets sent: 258689
 Heartbeat packets received: 258684
 Control link 1:
 Heartbeat packets sent: 258689
 Heartbeat packets received: 258684
Fabric link statistics:
 Probes sent: 258681
 Probes received: 258681
Services Synchronized:
 Service name RT0s sent RT0s received
 Translation context 0 0
 Incoming NAT 0 0
 Resource manager 0 0
 Session create 1 0
 Session close 1 0
 Session change 0 0
 Gate create 0 0
```

|                                 |   |   |
|---------------------------------|---|---|
| Session ageout refresh requests | 0 | 0 |
| Session ageout refresh replies  | 0 | 0 |
| IPSec VPN                       | 0 | 0 |
| Firewall user authentication    | 0 | 0 |
| MGCP ALG                        | 0 | 0 |
| H323 ALG                        | 0 | 0 |
| SIP ALG                         | 0 | 0 |
| SCCP ALG                        | 0 | 0 |
| PPTP ALG                        | 0 | 0 |
| RPC ALG                         | 0 | 0 |
| RTSP ALG                        | 0 | 0 |
| RAS ALG                         | 0 | 0 |
| MAC address learning            | 0 | 0 |
| GPRS GTP                        | 0 | 0 |

```
{primary:node1}
```

```
user@host> show chassis cluster statistics
```

```
Control link statistics:
```

```
Control link 0:
```

```
Heartbeat packets sent: 82371
```

```
Heartbeat packets received: 82321
```

```
Control link 1:
```

```
Heartbeat packets sent: 0
```

```
Heartbeat packets received: 0
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
  - Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
  - Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
  - Verifying a Chassis Cluster Configuration on page 869
  - Clearing Chassis Cluster Statistics on page 871
  - Verifying Chassis Cluster Failover Status on page 872

## Clearing Chassis Cluster Statistics

**Purpose** Clear displayed information about chassis cluster services and interfaces.

**Action** From the CLI, enter the **clear chassis cluster statistics** command:

```
{primary:node1}
```

```
user@host> clear chassis cluster statistics
```

```
Cleared control-plane statistics
```

```
Cleared data-plane statistics
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
  - Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
  - Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869

- Verifying Chassis Cluster Statistics on page 870
- Verifying Chassis Cluster Failover Status on page 872

## Verifying Chassis Cluster Failover Status

**Purpose** Display the failover status of a chassis cluster.

**Action** From the CLI, enter the **show chassis cluster status** command:

```
{primary:node1}
user@host> show chassis cluster status
Cluster ID: 3
```

| Node name                              | Priority | Status    | Preempt | Manual failover |
|----------------------------------------|----------|-----------|---------|-----------------|
| Redundancy-group: 0, Failover count: 1 |          |           |         |                 |
| node0                                  | 254      | primary   | no      | no              |
| node1                                  | 2        | secondary | no      | no              |
| Redundancy-group: 1, Failover count: 1 |          |           |         |                 |
| node0                                  | 254      | primary   | no      | no              |
| node1                                  | 1        | secondary | no      | no              |

```
{primary:node1}
user@host> show chassis cluster status
Cluster ID: 15
```

| Node                                     | Priority | Status  | Preempt | Manual failover |
|------------------------------------------|----------|---------|---------|-----------------|
| Redundancy group: 0 , Failover count: 5  |          |         |         |                 |
| node0                                    | 200      | primary | no      | no              |
| node1                                    | 0        | lost    | n/a     | n/a             |
| Redundancy group: 1 , Failover count: 41 |          |         |         |                 |
| node0                                    | 101      | primary | no      | no              |
| node1                                    | 0        | lost    | n/a     | n/a             |

```
{primary:node1}
user@host> show chassis cluster status
Cluster ID: 15
```

| Node                                     | Priority | Status      | Preempt | Manual failover |
|------------------------------------------|----------|-------------|---------|-----------------|
| Redundancy group: 0 , Failover count: 5  |          |             |         |                 |
| node0                                    | 200      | primary     | no      | no              |
| node1                                    | 0        | unavailable | n/a     | n/a             |
| Redundancy group: 1 , Failover count: 41 |          |             |         |                 |
| node0                                    | 101      | primary     | no      | no              |
| node1                                    | 0        | unavailable | n/a     | n/a             |

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
  - Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
  - Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
  - Verifying a Chassis Cluster Configuration on page 869



- Verifying Chassis Cluster Statistics on page 870
- Clearing Chassis Cluster Failover Status on page 873

## Clearing Chassis Cluster Failover Status

**Purpose** Clear the failover status of a chassis cluster.

**Action** From the CLI, enter the **clear chassis cluster failover-count** command:

```
{primary:node1}
user@host> clear chassis cluster failover-count
Cleared failover-count for all redundancy-groups
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Setting the Chassis Cluster Node ID and Cluster ID (CLI) on page 866
  - Example: Configuring the Chassis Cluster Management Interface (CLI) on page 867
  - Example: Configuring the Number of Redundant Ethernet Interfaces in a Chassis Cluster (CLI) on page 869
  - Verifying a Chassis Cluster Configuration on page 869
  - Verifying Chassis Cluster Statistics on page 870
  - Verifying Chassis Cluster Failover Status on page 872

## Chassis Cluster Upgrades ---

- Upgrading Each Device in a Chassis Cluster Separately on page 873
- Upgrading Both Devices in a Chassis Cluster Using a Low-Impact ISSU on page 874

### Upgrading Each Device in a Chassis Cluster Separately

Devices in a chassis cluster can be upgraded separately one at a time; some models allow one device after the other to be upgraded using failover and an in-service software upgrade (ISSU) to reduce the operational impact of the upgrade.

To upgrade each device in a chassis cluster separately:



**NOTE:** During this type of chassis cluster upgrade, a service disruption of about 3 to 5 minutes occurs.

1. Load the new image file on node 0.
2. Perform the image upgrade without rebooting the node by entering:  

```
user@host> request system software add image_name
```
3. Load the new image file on node 1.

4. Repeat Step 2.
5. Reboot both nodes simultaneously.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - [Upgrading Both Devices in a Chassis Cluster Using a Low-Impact ISSU on page 874](#)
  - [Disabling Chassis Cluster on page 877](#)
  - [Verifying a Chassis Cluster Configuration on page 869](#)
  - [Understanding Chassis Cluster Formation on page 796](#)

## Upgrading Both Devices in a Chassis Cluster Using a Low-Impact ISSU

- [Upgrading Both Devices in a Chassis Cluster Using an ISSU on page 874](#)
- [Rolling Back Devices in a Chassis Cluster After an ISSU on page 875](#)
- [Guarding Against Service Failure in a Chassis Cluster ISSU on page 875](#)
- [Enabling an Automatic Chassis Cluster Node Failback After an ISSU on page 876](#)
- [Troubleshooting Chassis Cluster ISSU Failures on page 876](#)
- [Deciphering Mismatched Control Link Statistics During a Chassis Cluster ISSU on page 876](#)

### Upgrading Both Devices in a Chassis Cluster Using an ISSU

For some platforms, devices in a chassis cluster can be upgraded without a service disruption using an in-service software upgrade (ISSU). The chassis cluster ISSU feature allows both devices in a cluster to be upgraded from supported Junos OS versions with a traffic impact similar to that of redundancy group failovers.



NOTE: ISSU does not support software downgrades.



NOTE: If you upgrade from a Junos OS version that supports only IPv4 to a version that supports both IPv4 and IPv6, the IPv4 traffic will continue to work during the upgrade process. If you upgrade from a Junos OS version that supports both IPv4 and IPv6 to a version that supports both IPv4 and IPv6, both the IPv4 and IPv6 traffic will continue to work during the upgrade process. Junos OS Release 10.2 and later releases support flow-based processing for IPv6 traffic. For more information, see “Enabling Flow-Based Processing for IPv6 Traffic” in the *Junos OS Security Configuration Guide*.

Before you begin, note the following:

- The ISSUs are available only for Junos OS Release 9.6 and later.
- Before starting an ISSU, you should fail over all redundancy groups so that they are all active on only one device. See “Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 814

- We recommend that routing protocols graceful restart be enabled prior to starting an ISSU.

Once all redundancy groups are active on one device, the upgrade is initiated by using a request command:

1. Fail over all redundancy groups to one device.
2. Start the ISSU by entering the following command:

```
user@host> request system software in-service-upgrade image_name reboot
```

If **reboot** is not included in the command, you need to manually reboot each device as the ISSU completes the software image update.

3. Wait for both devices to complete the upgrade, then verify that all policies, zones, redundancy groups, and other RTOs return to their correct states. Also verify that both devices in the cluster are running the new Junos OS build.



**NOTE:** During the upgrade, both devices might experience redundancy group failovers, but traffic is not disrupted. Each device validates the package and checks version compatibility before doing the upgrade. If the system finds that the new package is not version compatible with the currently installed version, the device refuses the upgrade or prompts you to take corrective action. Sometimes a single feature is not compatible, in which case the upgrade software prompts you to either abort the upgrade or turn off the feature before doing the upgrade.

This feature is available only through the command-line interface. See the “request system software in-service-upgrade” section of the *Junos OS CLI Reference*.

### Rolling Back Devices in a Chassis Cluster After an ISSU

If the ISSU fails to complete and only one device in the cluster has been upgraded, you can roll back to the previous configuration on that device alone by using the following commands on the upgraded device:

- **request chassis cluster in-service-upgrade abort**
- **request system software rollback**
- **request system reboot**

### Guarding Against Service Failure in a Chassis Cluster ISSU

The ISSU command has one option: **no-old-master-upgrade**. This option leaves the current master device in a nonupgraded state, which is a precaution against service failure. The **no-old-master-upgrade** option allows routing control to be quickly returned to the old master device if the newly upgraded device does not operate correctly.

Use of the **no-old-master-upgrade** option requires that you run a standard upgrade on the old master device after the ISSU is completed on the backup device.

If you use the **no-old-master-upgrade** option, when the backup device completes its upgrade and you are confident that the new build is operating as expected, then upgrade the old master as follows:

1. Run **request system software add *image\_name***.
2. Run **request chassis cluster in-service-upgrade abort** to stop the ISSU process.
3. Run **request system reboot**.

### Enabling an Automatic Chassis Cluster Node Failback After an ISSU

If you want redundancy groups to automatically return to node 0 as the primary after the ISSU is complete, you must set the redundancy group priority such that node 0 is primary and enable the preempt option. Note that this method works for all redundancy groups except redundancy group 0. You must manually fail over redundancy group 0. To set the redundancy group priority and enable the preempt option, see “Example: Configuring Chassis Cluster Redundancy Groups (CLI)” on page 803. To manually fail over a redundancy group, see “Initiating a Chassis Cluster Manual Redundancy Group Failover” on page 814.



**NOTE:** To upgrade node 0 and make it available in the chassis cluster, manually reboot node 0. Node 0 does not reboot automatically.

---

### Troubleshooting Chassis Cluster ISSU Failures

Certain circumstances might cause an ISSU attempt to fail. This section explains two of them.

- If you attempt to upgrade a device pair running a Junos OS image earlier than Release 9.6, the ISSU will fail without changing anything about either device in the cluster. Devices running Junos OS Releases earlier than 9.6 must be upgraded separately using individual device upgrade procedures.
- If the secondary device experiences a power-off condition before it boots up using the new image specified when the ISSU is initiated, when power is restored the newly upgraded device will still be waiting to end the ISSU. To end the ISSU on the secondary device, run **request chassis cluster in-service-upgrade abort** followed by **reboot** to abort the ISSU on that device.

### Deciphering Mismatched Control Link Statistics During a Chassis Cluster ISSU

When using dual control links (supported on the SRX5000 and SRX3000 lines only), mismatched control link statistics might be reported with the **show chassis cluster statistics** and **show chassis cluster control-plane statistics** commands while you run an ISSU with nodes on devices running different releases. (ISSUs are available in Junos OS Release 9.6 and later and dual control links are available in Junos OS Release 10.0 and later.) For example, assume that one node on a device is running Junos OS Release 9.6 and another node on a device is running Junos OS Release 10.0. In this example, a mismatch might occur because the latter device is sending heartbeats on both control links, but the other device is receiving heartbeats only on one control link.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- [Upgrading Each Device in a Chassis Cluster Separately on page 873](#)
- [Disabling Chassis Cluster on page 877](#)

## Disabling Chassis Cluster

---

To disable chassis cluster, enter the following command:

```
{primary:node1}
user@host> set chassis cluster disable reboot
Successfully disabled chassis cluster. Going to reboot now.
After the system reboots, the chassis cluster is disabled.
```

- Related Topics**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Upgrading Each Device in a Chassis Cluster Separately on page 873](#)
  - [Upgrading Both Devices in a Chassis Cluster Using a Low-Impact ISSU on page 874](#)
  - [Understanding What Happens When Chassis Cluster Is Enabled on page 845](#)
  - [Understanding Chassis Cluster Formation on page 796](#)

## Understanding Multicast Routing on a Chassis Cluster

---

Multicast routing support across nodes in a chassis cluster allows multicast protocols, such as Protocol Independent Multicast (PIM) versions 1 and 2, Internet Group Management Protocol (IGMP), Session Announcement Protocol (SAP), and Distance Vector Multicast Routing Protocol (DVMRP), to send traffic across interfaces in the cluster. Note, however, that the multicast protocols should not be enabled on the chassis management interface (**fxp0**) or on the fabric interfaces (**fab0** and **fab1**). Multicast sessions will be synched across the cluster and will be maintained during redundant group failovers. During failover, as with other types of traffic, there might be some multicast packet loss.

Multicast data forwarding in a chassis cluster uses the incoming interface to determine whether or not the session remains active. Packets will be forwarded to the peer node if a leaf session's outgoing interface is on the peer instead of on the incoming interface's node. Multicast routing on a chassis cluster supports tunnels for both incoming and outgoing interfaces.

Multicast configuration on a chassis cluster is the same as multicast configuration on a standalone device (see the “Configuring a Multicast Network” chapter of the *Junos OS Routing Protocols and Policies Configuration Guide for Security Devices*).

- Related Topics**
- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
  - [Chassis Cluster Overview on page 795](#)
  - [Understanding Chassis Cluster Formation on page 796](#)

## Asymmetric Chassis Cluster Deployment

- Understanding Asymmetric Routing Chassis Cluster Deployment on page 878
- Example: Configuring an Asymmetric Chassis Cluster Pair (CLI) on page 880
- Example: Configuring an Asymmetric Chassis Cluster Pair (J-Web) on page 881

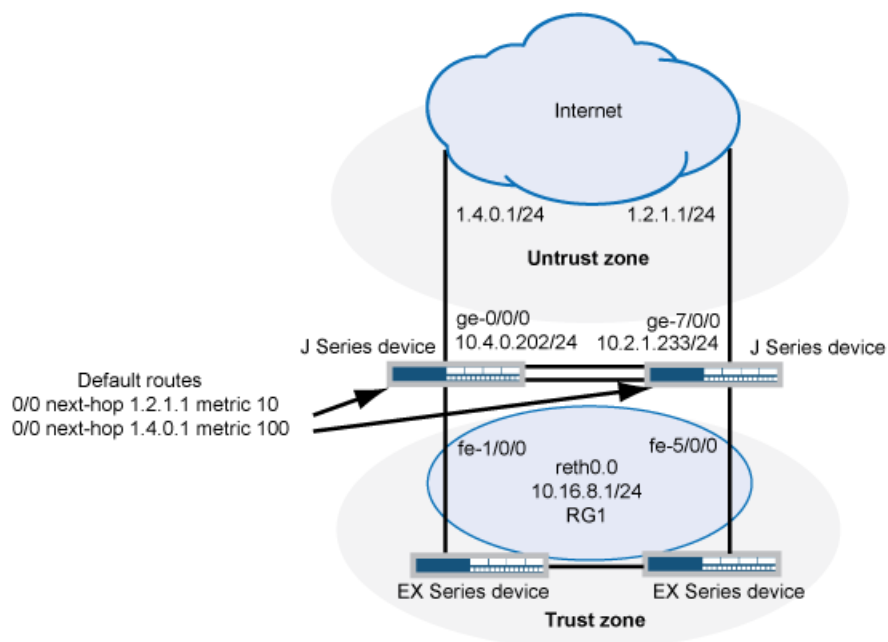
### Understanding Asymmetric Routing Chassis Cluster Deployment

In this case, chassis cluster makes use of its asymmetric routing capability (see Figure 103 on page 878). Traffic received by a node is matched against that node's session table. The result of this lookup determines whether or not that node should process the packet or forward it to the other node over the fabric link. Sessions are anchored on the egress node for the first packet that created the session. If traffic is received on the node in which the session is not anchored, those packets are forwarded over the fabric link to the node where the session is anchored.



**NOTE:** The anchor node for the session can change if there are changes in routing during the session.

**Figure 103: Asymmetric Routing Chassis Cluster Scenario (J Series Devices)**



In this scenario, two Internet connections are used, with one being preferred. The connection to the trust zone is done by using a redundant Ethernet interface to provide LAN redundancy for the devices in the trust zone. This scenario describes two failover

cases in which sessions originate in the trust zone with a destination of the Internet (untrust zone).

- Understanding Failures in the Trust Zone Redundant Ethernet Interface on page 879
- Understanding Failures in the Untrust Zone Interfaces on page 879

## Understanding Failures in the Trust Zone Redundant Ethernet Interface

Under normal operating conditions, traffic flows from the trust zone interface **fe-1/0/0**, belonging to **reth0.0**, to the Internet. Because the primary Internet connection is on node 0, sessions are both created in node 0 and synced to node 1. However, sessions are only active on node 0.

A failure in interface **fe-1/0/0** triggers a failover of the redundancy group, causing interface **fe-5/0/0** in node 1 to become active. After the failover, traffic arrives at node 1. After session lookup, the traffic is sent to node 0 because the session is active on this node. Node 0 then processes the traffic and forwards it to the Internet. The return traffic follows a similar process. The traffic arrives at node 0 and gets processed for security purposes—for example, antispam scanning, antivirus scanning, and application of security policies—on node 0 because the session is anchored to node 0. The packet is then sent to node 1 through the fabric interface for egress processing and eventual transmission out of node 1 through interface **fe-5/0/0**.

## Understanding Failures in the Untrust Zone Interfaces

In this case, sessions are migrated from node to node. Under normal operating conditions, traffic is processed by only node 0. A failure of interface **ge-0/0/0** on node 0 causes a change in the routing table, so that it now points to interface **ge-7/0/0** in node 1. After the failure, sessions in node 0 become inactive, and the passive sessions in node 1 become active. Traffic arriving from the trust zone is still received on interface **fe-1/0/0**, but is forwarded to node 1 for processing. After traffic is processed in node 1, it is forwarded to the Internet through interface **ge-7/0/0**.

In this chassis cluster configuration, redundancy group 1 is used to control the redundant Ethernet interface connected to the trust zone. As configured in this scenario, redundancy group 1 fails over only if interface **fe-1/0/0** or **fe-5/0/0** fails, but not if the interfaces connected to the Internet fail. Optionally, the configuration could be modified to permit redundancy group 1 to monitor all interfaces connected to the Internet and fail over if an Internet link were to fail. So, for example, the configuration can allow redundancy group 1 to monitor **ge-0/0/0** and make **fe-5/0/0** active for **reth0** if the **ge-0/0/0** Internet link fails. (This option is not described in the following configuration examples.)

### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Example: Configuring an Asymmetric Chassis Cluster Pair (CLI) on page 880
- Example: Configuring an Asymmetric Chassis Cluster Pair (J-Web) on page 881
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- Understanding Chassis Cluster Formation on page 796

## Example: Configuring an Asymmetric Chassis Cluster Pair (CLI)



NOTE: First, do basic chassis cluster and management interfaces setup.

1. Configure the fabric interface.

```
{primary:node1}
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/1
{primary:node1}
user@host# set interfaces fab1 fabric-options member-interfaces ge-7/0/1
```

2. Configure the redundancy groups.

```
{primary:node1}
user@host# set chassis cluster reth-count 1
user@host# set chassis cluster heartbeat-interval 1000
{primary:node1}
user@host# set chassis cluster heartbeat-threshold 3
{primary:node1}
user@host# set chassis cluster node 0
{primary:node1}
user@host# set chassis cluster node 1
{primary:node1}
user@host# set chassis cluster redundancy-group 1 node 0 priority 100
{primary:node1}
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-1/0/0
weight 255
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-5/0/0
weight 255
```

3. Configure the redundant Ethernet interfaces.

```
{primary:node1}
user@host# set interfaces ge-0/0/0 unit 0 family inet address 1.4.0.202/24
{primary:node1}
user@host# set interfaces fe-1/0/0 fastether-options redundant-parent reth0
{primary:node1}
user@host# set interfaces ge-7/0/0 unit 0 family inet address 1.2.1.233/24
{primary:node1}
user@host# set interfaces fe-5/0/0 fastether-options redundant-parent reth0
{primary:node1}
user@host# set interfaces reth0 unit 0 family inet address 10.16.8.1/24
```

4. Configure the static routes (one to each ISP, with preferred route through ge-0/0/0).

```
{primary:node1}
user@host# set routing-options static route 0.0.0.0/0 qualified-next-hop 1.4.0.1
metric 10
{primary:node1}
user@host# set routing-options static route 0.0.0.0/0 qualified-next-hop 1.2.1.1
metric 100
```

5. Configure the security zones.



```
{primary:node1}
user@host# set security zones security-zone Untrust interfaces ge-0/0/0.0
{primary:node1}
user@host# set security zones security-zone Untrust interfaces ge-7/0/0.0
```

6. Configure the security policies.

```
{primary:node1}
user@host# set security policies from-zone Trust to-zone Untrust policy ANY match
source-address any
{primary:node1}
user@host# set security policies from-zone Trust to-zone Untrust policy ANY match
destination-address any
{primary:node1}
user@host# set security policies from-zone Trust to-zone Untrust policy ANY match
application any
{primary:node1}
user@host# set security policies from-zone Trust to-zone Untrust policy ANY then
permit
```

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Asymmetric Routing Chassis Cluster Deployment on page 878
- Example: Configuring an Asymmetric Chassis Cluster Pair (J-Web) on page 881
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- Understanding Chassis Cluster Formation on page 796

### Example: Configuring an Asymmetric Chassis Cluster Pair (J-Web)



**NOTE:** First, enable clustering and do management interfaces setup.

1. Configure the fabric interface.

See Step 1 in “Example: Configuring an Asymmetric Chassis Cluster Pair (CLI)” on page 880.

2. Configure the redundancy groups.

- Select **Configure>System Properties>Chassis Cluster**.
- Enter the following information, and then click **Apply**:

Redundant ether-Interface Count: 1

Heartbeat Interval: 1000

Heartbeat Threshold: 3

Nodes: 0

Group Number: 1

Priorities: 100

- Enter the following information, and then click **Apply**:
  - Nodes: 1
  - Group Number: 1
  - Priorities: 1
  - Interface Monitor—Interface: **fe-1/0/0**
  - Interface Monitor—Weight: **255**
  - Interface Monitor—Interface: **fe-5/0/0**
  - Interface Monitor—Weight: **255**
- 3. Configure the redundant Ethernet interfaces.
  - Select **Configure>Interfaces**.
  - Select **fe-1/0/0**.
  - Enter **reth0** in the Redundant Parent box.
  - Click **Apply**.
  - Select **fe-5/0/0**.
  - Enter **reth0** in the Redundant Parent box.
  - Click **Apply**.
  - See Step 3 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 884.
- 4. Configure the static routes (one to each ISP, with preferred route through **ge-0/0/0**).
  - Select **Configure>Routing>Static Routing**.
  - Click **Add**.
  - Enter the following information, and then click **Apply**:
    - Static Route Address: **0.0.0.0/0**
    - Next-Hop Addresses: **1.4.0.1, 1.2.1.1**
- 5. Configure the security zones.
  - See Step 5 in “Example: Configuring an Asymmetric Chassis Cluster Pair (CLI)” on page 880.
- 6. Configure the security policies.
  - See Step 6 in “Example: Configuring an Asymmetric Chassis Cluster Pair (CLI)” on page 880.
- 7. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Asymmetric Routing Chassis Cluster Deployment on page 878
  - Example: Configuring an Asymmetric Chassis Cluster Pair (CLI) on page 880
  - Understanding What Happens When Chassis Cluster Is Enabled on page 845
  - Understanding Chassis Cluster Formation on page 796

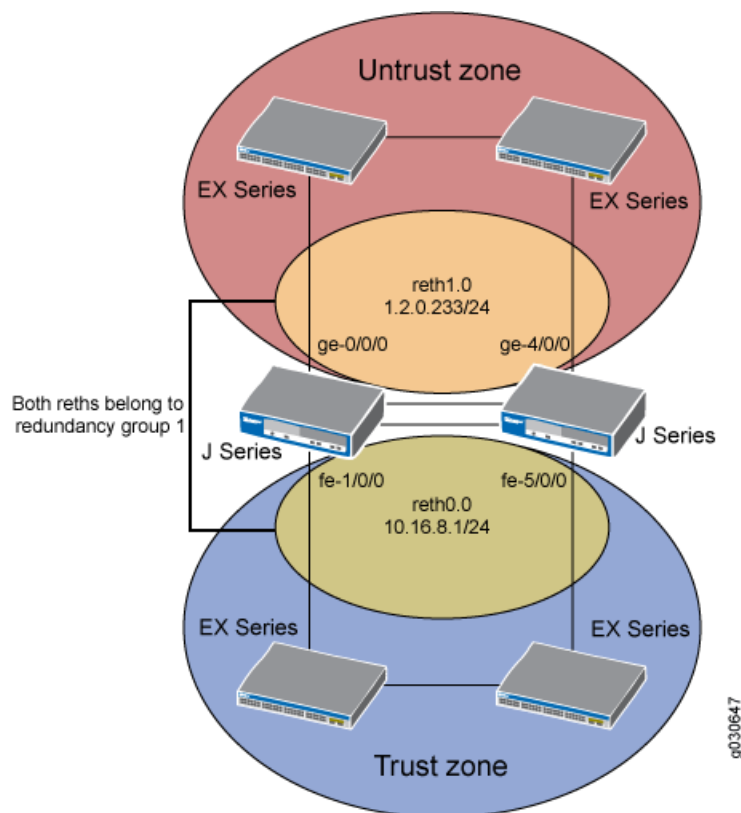
## Active/Passive Chassis Cluster Deployment (J Series Devices)

- Understanding Active/Passive Chassis Cluster Deployment on page 883
- Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) on page 884
- Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) on page 886

### Understanding Active/Passive Chassis Cluster Deployment

In this case, a single device in the cluster is used to route all traffic while the other device is used only in the event of a failure (see Figure 104 on page 883). When a failure occurs, the backup device becomes master and controls all forwarding.

**Figure 104: Active/Passive Chassis Cluster Scenario (J Series Devices)**



An active/passive chassis cluster can be achieved by using redundant Ethernet interfaces (reths) that are all assigned to the same redundancy group. If any of the interfaces in an

active group in a node fails, the group is declared inactive and all the interfaces in the group fail over to the other node.

This configuration minimizes the traffic over the fabric link because only one node in the cluster forwards traffic at any given time.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) on page 884
  - Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) on page 886
  - Understanding What Happens When Chassis Cluster Is Enabled on page 845
  - Understanding Chassis Cluster Formation on page 796

### Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)

1. Enable clustering.

In node 0:

```
user@host> set chassis cluster cluster-id 1 node 0
warning: A reboot is required for chassis cluster to be enabled
```

In node 1:

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
Successfully enabled chassis cluster. Going to reboot now.
```

2. Configure the management interface.

In a cluster, the configuration is shared among the cluster members. Member-specific configurations (such as the IP address of the management port of each member) are entered using configuration groups.

```
{primary:node1}
user@host# set groups node0 system host-name J2320-A
{primary:node1}
user@host# set groups node0 interfaces fxp0 unit 0 family inet address
192.168.3.110/24
{primary:node1}
user@host# set groups node1 system host-name J2320-B
{primary:node1}
user@host# set groups node1 interfaces fxp0 unit 0 family inet address
192.168.3.111/24
{primary:node1}
user@host# set apply-groups "${node}"
```

3. Configure the fabric interface.

```
{primary:node1}
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/1
{primary:node1}
user@host# set interfaces fab1 fabric-options member-interfaces ge-4/0/1
```

4. Configure the redundancy groups.

```
{primary:node1}
user@host# set chassis cluster reth-count 2
```

```

{primary:node1}
user@host# set chassis cluster heartbeat-interval 1000
{primary:node1}
user@host# set chassis cluster heartbeat-threshold 3
{primary:node1}
user@host# set chassis cluster node 0
{primary:node1}
user@host# set chassis cluster node 1
{primary:node1}
user@host# set chassis cluster redundancy-group 0 node 0 priority 100
{primary:node1}
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
{primary:node1}
user@host# set chassis cluster redundancy-group 1 node 0 priority 100
{primary:node1}
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-1/0/0
weight 255
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-5/0/0
weight 255
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-0/0/0
weight 255
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor ge-4/0/0
weight 255

```

5. Configure the redundant Ethernet interfaces.

```

{primary:node1}
user@host# set interfaces ge-0/0/0 gigether-options redundant-parent reth1
{primary:node1}
user@host# set interfaces ge-4/0/0 gigether-options redundant-parent reth1
{primary:node1}
user@host# set interfaces fe-1/0/0 fastether-options redundant-parent reth0
{primary:node1}
user@host# set interfaces fe-5/0/0 fastether-options redundant-parent reth0
{primary:node1}
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
{primary:node1}
user@host# set interfaces reth0 unit 0 family inet address 10.16.8.1/24
{primary:node1}
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
{primary:node1}
user@host# set interfaces reth1 unit 0 family inet address 1.2.0.233/24

```

6. Configure the security zones.

```

{primary:node1}
user@host# set security zones security-zone Untrust interfaces reth1.0
{primary:node1}
user@host# set security zones security-zone Trust interfaces reth0.0

```

7. Configure the security policies.

```

{primary:node1}

```

```
user@host# set security policies from-zone Trust to-zone Untrust policy ANY match
source-address any
{primary:node1}
user@host# set security policies from-zone Trust to-zone Untrust policy ANY match
destination-address any
{primary:node1}
user@host# set security policies from-zone Trust to-zone Untrust policy ANY match
application any
{primary:node1}
user@host# set security policies from-zone Trust to-zone Untrust policy ANY then
permit
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Active/Passive Chassis Cluster Deployment on page 883
  - Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web) on page 886
  - Understanding What Happens When Chassis Cluster Is Enabled on page 845
  - Understanding Chassis Cluster Formation on page 796

### Example: Configuring an Active/Passive Chassis Cluster Pair (J-Web)

1. Enable clustering. See Step 1 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 884.
2. Configure the management interface. See Step 2 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 884.
3. Configure the fabric interface. See Step 3 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 884.
4. Configure the redundancy groups.
  - Select **Configure>System Properties>Chassis Cluster**.
  - Enter the following information, and then click **Apply**:  
Redundant ether-Interface Count: **2**  
Heartbeat Interval: **1000**  
Heartbeat Threshold: **3**  
Nodes: **0**  
Group Number: **0**  
Priorities: **100**
  - Enter the following information, and then click **Apply**:  
Nodes: **0**  
Group Number: **1**  
Priorities: **1**

- Enter the following information, and then click **Apply**:
    - Nodes: 1
    - Group Number: 0
    - Priorities: 100
  - Enter the following information, and then click **Apply**:
    - Nodes: 1
    - Group Number: 1
    - Priorities: 1
    - Interface Monitor—Interface: **fe-1/0/0**
    - Interface Monitor—Weight: 255
    - Interface Monitor—Interface: **fe-5/0/0**
    - Interface Monitor—Weight: 255
    - Interface Monitor—Interface: **ge-0/0/0**
    - Interface Monitor—Weight: 255
    - Interface Monitor—Interface: **ge-4/0/0**
    - Interface Monitor—Weight: 255
5. Configure the redundant Ethernet interfaces.
- Select **Configure>System Properties>Chassis Cluster**.
  - Select **ge-0/0/0**.
  - Enter **reth1** in the Redundant Parent box.
  - Click **Apply**.
  - Select **ge-4/0/0**.
  - Enter **reth1** in the Redundant Parent box.
  - Click **Apply**.
  - Select **fe-1/0/0**.
  - Enter **reth0** in the Redundant Parent box.
  - Click **Apply**.
  - Select **fe-5/0/0**.

- Enter **reth0** in the Redundant Parent box.
  - Click **Apply**.
  - See Step 5 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 884 for the last four configuration settings.
6. Configure the security zones. See Step 6 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 884.
  7. Configure the security policies. See Step 7 in “Example: Configuring an Active/Passive Chassis Cluster Pair (CLI)” on page 884.
  8. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

**Related Topics**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Active/Passive Chassis Cluster Deployment on page 883
- Example: Configuring an Active/Passive Chassis Cluster Pair (CLI) on page 884
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- Understanding Chassis Cluster Formation on page 796

---

## Active/Passive Chassis Cluster Deployment (SRX Series Devices)

---

- Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster on page 888
- Example: Configuring an SRX Series Services Gateway for the High-End as a Chassis Cluster on page 902

### Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster

This example shows how to set up chassis clustering on an SRX Series for the branch device.

- Requirements on page 888
- Overview on page 889
- Configuration on page 893
- Verification on page 899

#### Requirements

Before you begin:



- Disable switching. Layer 2 Ethernet switching is not supported in chassis cluster mode. See “Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering” on page 860.
- Physically connect the two devices and ensure that they are the same models. For example, on the SRX210 Services Gateway, connect **fe-0/0/7** on node 0 to **fe-0/0/7** on node 1.
- Set the two devices to cluster mode and reboot the devices. You must enter the following operational mode commands on both devices, for example:

- On node 0:

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```

- On node 1:

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

The cluster-id is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster-id is 0 through 15 and setting it to 0 is equivalent to disabling cluster mode.

- After clustering occurs for the devices, continuing with the SRX210 Services Gateway example, the **fe-0/0/7** interface on node 1 changes to **fe-2/0/7**. After the reboot, the following interfaces are assigned and repurposed to form a cluster:
  - **fe-0/0/6** becomes **fxp0** and is used for individual management of the chassis cluster.
  - **fe-0/0/7** becomes **fxp1** is used as the control link within the chassis cluster.
  - The other interfaces are also renamed on the secondary device. For example, the **ge-0/0/0** interface is renamed **ge-2/0/0** on node 1 on the secondary device.

See “Node Interfaces on Active SRX Series Chassis Clusters” on page 846 for complete mapping of the SRX Series devices.



**NOTE:** The ports used for the control link, **fe-0/0/7**, must be connected with a cable. A switch cannot be used to connect the control link. You must also decide which port to use as the third link to connect the devices and use as the fabric link between the devices. This port can be any available Gigabit Ethernet or Fast Ethernet port other than **fe-0/0/6** and **fe-0/0/7**.

From this point forward, configuration of the cluster is synchronized between the node members and the two separate devices function as one device.

### Overview

This example shows how to set up chassis clustering on an SRX Series for the branch device. The following services gateways for the branch are supported:

- SRX100
- SRX210

- SRX240
- SRX650

Depending on the device used, node 1 rennumbers its interfaces by adding the total number of system FPCs to the original FPC number of the interface. See Table 82 on page 890 for interface renumbering on the SRX Series device.

**Table 82: SRX Series Services Gateways Interface Renumbering**

| SRX Series Services Gateway | Control Link Name | Renumbering Constant | Node 0 Interface Name | Node 1 Interface Name |
|-----------------------------|-------------------|----------------------|-----------------------|-----------------------|
| SRX100                      | fe-0/0/7          | 1                    | fe-0/0/0              | fe-1/0/0              |
| SRX210                      | fe-0/0/7          | 2                    | ge-0/0/0              | ge-2/0/0              |
| SRX240                      | ge-0/0/1          | 5                    | ge-0/0/0              | ge-5/0/0              |
| SRX650                      | ge-0/0/1          | 9                    | ge-0/0/0              | ge-9/0/0              |

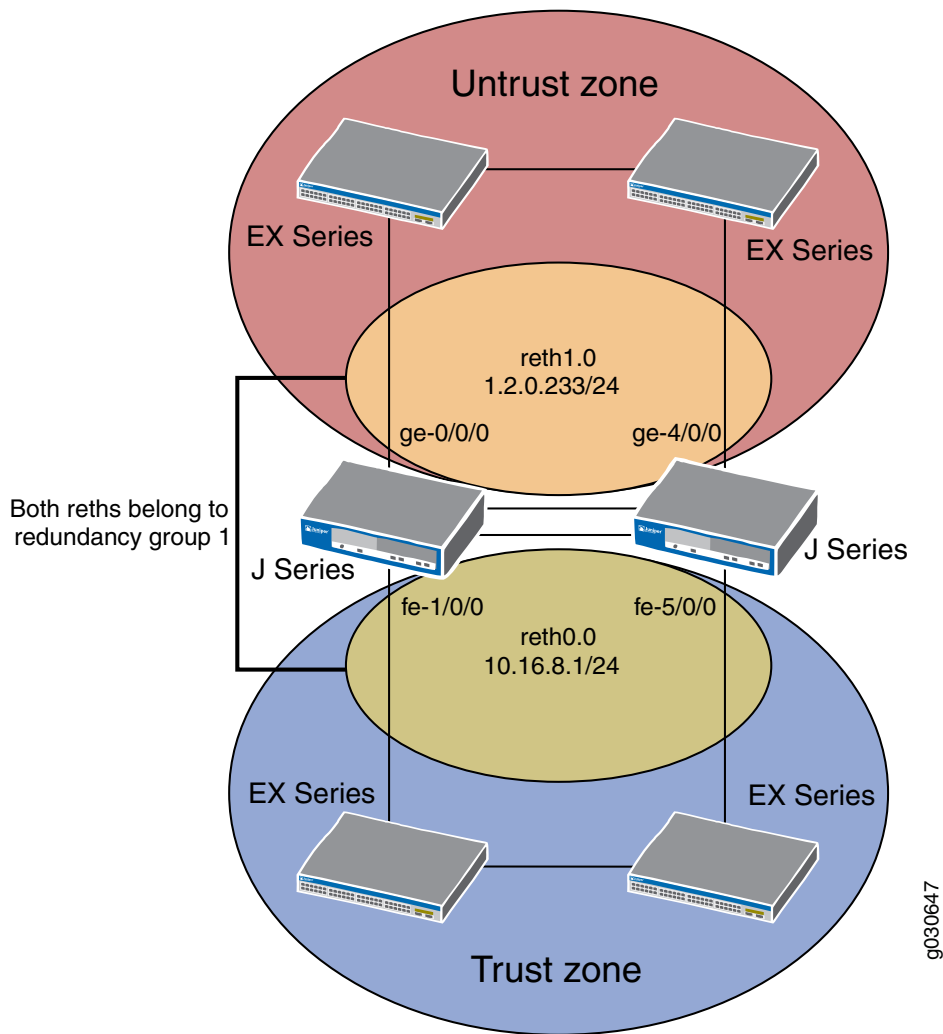
After clustering is enabled, the system creates fxp0, fxp1, and fab interfaces. Depending on the device, the fxp0 and fxp1 interfaces that are mapped to a physical interface are not user defined. However, the fab interface is user defined. see Table 83 on page 890 for mapping of the fxp0 and fxp1 interfaces on the SRX Series devices.

**Table 83: SRX Series Services Gateways fxp0 and fxp1 Interfaces Mapping**

| SRX Series Services Gateway | fxp0 Interface | fxp1 Interface | fab Interface |
|-----------------------------|----------------|----------------|---------------|
| SRX100                      | fe-0/0/6       | fe-0/0/7       | user defined  |
| SRX210                      | ge-0/0/0       | fe-0/0/7       | user defined  |
| SRX240                      | ge-0/0/0       | ge-0/0/1       | user defined  |
| SRX650                      | ge-0/0/0       | ge-0/0/1       | user defined  |

Figure 105 on page 891 shows the topology used in this example.

Figure 105: SRX Series for the Branch Topology Example



g030647

### Configuration

**CLI Quick Configuration** To quickly configure a chassis cluster on an SRX210 Services Gateway, copy the following commands and paste them into the CLI:

On {primary:node0}

```
[edit]
set groups node0 system host-name srx-node0
set groups node0 interfaces fxp0 unit 0 family inet address 192.16.35.46/24
set groups node1 system host-name srx-node1
set groups node1 interfaces fxp0 unit 0 family inet address 192.16.35.47/24
set groups node0 system backup-router <backup next-hop from fxp0> destination
<management network/mask>
set groups node1 system backup-router <backup next-hop from fxp0> destination
<management network/mask>
set apply-groups "${node}"
set interfaces fab0 fabric-options member-interfaces ge-0/0/1
set interfaces fab1 fabric-options member-interfaces ge-2/0/1
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 1
set chassis cluster redundancy-group 1 interface-monitor fe-0/0/3 weight 255
set chassis cluster redundancy-group 1 interface-monitor fe-0/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor fe-2/0/3 weight 255
set chassis cluster redundancy-group 1 interface-monitor fe-2/0/2 weight 255
set chassis cluster reth-count 2
set interfaces fe-0/0/2 fastether-options redundant-parent reth1
set interfaces fe-2/0/2 fastether-options redundant-parent reth1
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 1.2.0.233/24
set interfaces fe-0/0/3 fastether-options redundant-parent reth0
set interfaces fe-2/0/3 fastether-options redundant-parent reth0
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.16.8.1/24
set security zones security-zone Untrust interfaces reth1.0
set security zones security-zone Trust interfaces reth0.0
```

If you are configuring an SRX Series for the branch device other than the SRX210 device, see Table 84 on page 893 for command and interface settings for your device and substitute these commands into your CLI.

**Table 84: SRX Series Services Gateways for the Branch Interface Settings**

| Command                                                    | SRX100   | SRX210   | SRX240   | SRX650   |
|------------------------------------------------------------|----------|----------|----------|----------|
| set interfaces fab0<br>fabric-options<br>member-interfaces | fe-0/0/1 | ge-0/0/1 | ge-0/0/2 | ge-0/0/2 |
| set interfaces fab1<br>fabric-options<br>member-interfaces | fe-1/0/1 | ge-2/0/1 | ge-5/0/2 | ge-9/0/2 |

Table 84: SRX Series Services Gateways for the Branch Interface Settings (*continued*)

| Command                                                        | SRX100                                                  | SRX210                                                  | SRX240                                                 | SRX650                                                  |
|----------------------------------------------------------------|---------------------------------------------------------|---------------------------------------------------------|--------------------------------------------------------|---------------------------------------------------------|
| set chassis cluster<br>redundancy-group 1<br>interface-monitor | fe-0/0/0 weight 255                                     | fe-0/0/3 weight 255                                     | ge-0/0/5 weight 255                                    | ge-1/0/0 weight 255                                     |
| set chassis cluster<br>redundancy-group 1<br>interface-monitor | fe-0/0/2 weight 255                                     | fe-0/0/2 weight 255                                     | ge-5/0/5 weight 255                                    | ge-10/0/0 weight 255                                    |
| set chassis cluster<br>redundancy-group 1<br>interface-monitor | fe-1/0/0 weight 255                                     | fe-2/0/3 weight 255                                     | ge-0/0/6 weight 255                                    | ge-1/0/1 weight 255                                     |
| set chassis cluster<br>redundancy-group 1<br>interface-monitor | fe-1/0/2 weight 255                                     | fe-2/0/2 weight 255                                     | ge-5/0/6 weight 255                                    | ge-10/0/1 weight 255                                    |
| set interfaces                                                 | fe-0/0/2<br>fastether-options<br>redundant-parent reth1 | fe-0/0/2<br>fastether-options<br>redundant-parent reth1 | ge-0/0/5<br>gigether-options<br>redundant-parent reth1 | ge-1/0/0<br>gigether-options<br>redundant-parent reth1  |
| set interfaces                                                 | fe-1/0/2<br>fastether-options<br>redundant-parent reth1 | fe-2/0/2<br>fastether-options<br>redundant-parent reth1 | ge-5/0/5<br>gigether-options<br>redundant-parent reth1 | ge-10/0/0<br>gigether-options<br>redundant-parent reth1 |
| set interfaces                                                 | fe-0/0/0<br>fastether-options<br>redundant-parent reth0 | fe-0/0/3<br>fastether-options<br>redundant-parent reth0 | ge-0/0/6<br>gigether-options<br>redundant-parent reth0 | ge-1/0/1<br>gigether-options<br>redundant-parent reth0  |
| set interfaces                                                 | fe-1/0/0<br>fastether-options<br>redundant-parent reth0 | fe-2/0/3<br>fastether-options<br>redundant-parent reth0 | ge-5/0/6<br>gigether-options<br>redundant-parent reth0 | ge-10/0/1<br>gigether-options<br>redundant-parent reth0 |

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a chassis cluster on an SRX Series for the branch device:



**NOTE:** Perform Steps 1 through 5 on the primary device (node 0). They are automatically copied over to the secondary device (node 1) when you execute a `commit` command. The configurations are synchronized because the control link and fab link interfaces are activated. To verify the configurations, use the `show interface terse` command and review the output.

1. Set up hostnames and management IP addresses for each device using configuration groups. These configurations are specific to each device and are unique to its specific node.

```
user@host# set groups node0 system host-name srx-node0
user@host# set groups node0 interfaces fxp0 unit 0 family inet address
192.16.35.46/24
user@host# set groups node1 system host-name srx-node1
user@host# set groups node1 interfaces fxp0 unit 0 family inet address
192.16.35.47/24
```

Set the default route and backup router for each node.

```
set groups node0 system backup-router <backup next-hop from fxp0> destination
<management network/mask>
set groups node1 system backup-router <backup next-hop from fxp0> destination
<management network/mask>
```

Set the **apply-group** command so that the individual configurations for each node set by the previous commands are applied only to that node.

```
user@host# set apply-groups "${node}"
```

2. Define the interfaces used for the fab connection (data plane links for RTO sync) by using physical ports **ge-0/0/1** from each node. These interfaces must be connected back-to-back, or through a Layer 2 infrastructure.

```
user@host# set interfaces fab0 fabric-options member-interfaces ge-0/0/1
user@host# set interfaces fab1 fabric-options member-interfaces ge-2/0/1
```

3. Set up redundancy group 0 for the Routing Engine failover properties, and set up redundancy group 1 (all interfaces are in one redundancy group in this example) to define the failover properties for the redundant Ethernet interfaces.

```
user@host# set chassis cluster redundancy-group 0 node 0 priority 100
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
user@host# set chassis cluster redundancy-group 1 node 0 priority 100
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
```

4. Set up interface monitoring to monitor the health of the interfaces and trigger redundancy group failover.



NOTE: We do not recommend Interface monitoring for redundancy group 0 because it causes the control plane to switch from one node to another node in case interface flap occurs.

```
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-0/0/3
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-0/0/2
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-2/0/3
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor fe-2/0/2
weight 255
```



NOTE: Interface failover only occurs after the weight reaches 0.

5. Set up the redundant Ethernet (reth) interfaces and assign the redundant interface to a zone.

```
user@host# set chassis cluster reth-count 2
user@host# set interfaces fe-0/0/2 fastether-options redundant-parent reth1
user@host# set interfaces fe-2/0/2 fastether-options redundant-parent reth1
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 1.2.0.233/24
user@host# set interfaces fe-0/0/3 fastether-options redundant-parent reth0
user@host# set interfaces fe-2/0/3 fastether-options redundant-parent reth0
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 10.16.8.1/24
user@host# set security zones security-zone Untrust interfaces reth1.0
user@host# set security zones security-zone Trust interfaces reth0.0
```

**Results** From operational mode, confirm your configuration by entering the **show configuration** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
> show configuration
version x.xx.x;
groups {
 node0 {
 system {
 host-name SRX210-1;
 backup-router 10.100.22.1 destination 66.129.243.0/24;
 }
 interfaces {
 fxp0 {
 unit 0 {
```



```

 family inet {
 address 192.16.35.46/24;
 }
 }
}
}
node1 {
 system {
 host-name SRX210-2;
 backup-router 10.100.21.1 destination 66.129.243.0/24; }
 interfaces {
 fxp0 {
 unit 0 {
 family inet {
 address 192.16.35.47/24;
 }
 }
 }
 }
}
}
apply-groups "${node}";
chassis {
 cluster {
 reth-count 2;
 redundancy-group 0 {
 node 0 priority 100;
 node 1 priority 1;
 }
 redundancy-group 1 {
 node 0 priority 100;
 node 1 priority 1;
 interface-monitor {
 fe-0/0/3 weight 255;
 fe-0/0/2 weight 255;
 fe-2/0/2 weight 255;
 fe-2/0/3 weight 255;
 }
 }
 }
}
interfaces {
 fe-0/0/2 {
 fastether-options {
 redundant-parent reth1;
 }
 unit 0 {
 family inet {
 address 2.2.2.2/30;
 }
 }
 }
 fe-0/0/3 {
 fastether-options {
 redundant-parent reth0;
 }
 }
 fe-2/0/2 {
 fastether-options {
 redundant-parent reth1;
 }
 }
}

```

```
 }
 }
 fe-2/0/3 {
 fastether-options {
 redundant-parent reth0;
 }
 }
 fab0 {
 fabric-options {
 member-interfaces {
 ge-0/0/1;
 }
 }
 }
 fab1 {
 fabric-options {
 member-interfaces {
 ge-2/0/1;
 }
 }
 }
 reth0 {
 redundant-ether-options {
 redundancy-group 1;
 }
 unit 0 {
 family inet {
 address 10.16.8.1/24;
 }
 }
 }
 reth1 {
 redundant-ether-options {
 redundancy-group 1;
 }
 unit 0 {
 family inet {
 address 1.2.0.233/24;
 }
 }
 }
}
...
security {
 zones {
 security-zone Untrust {
 interfaces {
 reth1.0;
 }
 }
 security-zone Trust {
 interfaces {
 reth0.0;
 }
 }
 }
 policies {
 from-zone Trust to-zone Untrust {
 policy 1 {
 match {
 source-address any;
 }
 }
 }
 }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

To confirm that the configuration is working properly, perform these tasks:

- ## Verifying Chassis Cluster Status

**Action** From operational mode, enter the **show chassis cluster status** command.

## Verifying Chassis Cluster Interfaces

**Action** From operational mode, enter the **show chassis cluster interfaces** command.

```

Redundant-ethernet Information:
 Name Status Redundancy-group
 reth0 Up 1
 reth1 Up 1

```

**Interface Monitoring:**

| Interface | Weight | Status | Redundancy-group |
|-----------|--------|--------|------------------|
| fe-2/0/3  | 255    | Up     | 1                |
| fe-2/0/2  | 255    | Up     | 1                |
| fe-0/0/2  | 255    | Up     | 1                |
| fe-0/0/3  | 255    | Up     | 1                |

**Verifying Chassis Cluster Statistics**

**Purpose** Verify information about the statistics of the different objects being synchronized, the fabric and control interface hellos, and the status of the monitored interfaces in the cluster.

**Action** From operational mode, enter the **show chassis cluster statistics** command.

```
{primary:node0}
user@host> show chassis cluster statistics
```

Control link statistics:

Control link 0:

Heartbeat packets sent: 2276

Heartbeat packets received: 2280

Heartbeat packets errors: 0

Fabric link statistics:

Probes sent: 2272

Probes received: 597

Probe errors: 0

Services Synchronized:

| Service name                    | RTOs sent | RTOs received |
|---------------------------------|-----------|---------------|
| Translation context             | 0         | 0             |
| Incoming NAT                    | 0         | 0             |
| Resource manager                | 6         | 0             |
| Session create                  | 161       | 0             |
| Session close                   | 148       | 0             |
| Session change                  | 0         | 0             |
| Gate create                     | 0         | 0             |
| Session ageout refresh requests | 0         | 0             |
| Session ageout refresh replies  | 0         | 0             |
| IPSec VPN                       | 0         | 0             |
| Firewall user authentication    | 0         | 0             |
| MGCP ALG                        | 0         | 0             |
| H323 ALG                        | 0         | 0             |
| SIP ALG                         | 0         | 0             |
| SCCP ALG                        | 0         | 0             |
| PPTP ALG                        | 0         | 0             |
| RPC ALG                         | 0         | 0             |
| RTSP ALG                        | 0         | 0             |
| RAS ALG                         | 0         | 0             |
| MAC address learning            | 0         | 0             |
| GPRS GTP                        | 0         | 0             |

**Verifying Chassis Cluster Control Plane Statistics**

**Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

**Action** From operational mode, enter the **show chassis cluster control-plane statistics** command.

```
{primary:node0}
```

```
user@host> show chassis cluster control-plane statistics
```

```
Control link statistics:
```

```
Control link 0:
```

```
Heartbeat packets sent: 2294
```

```
Heartbeat packets received: 2298
```

```
Heartbeat packets errors: 0
```

```
Fabric link statistics:
```

```
Probes sent: 2290
```

```
Probes received: 615
```

```
Probe errors: 0
```

### Verifying Chassis Cluster Data Plane Statistics

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster data-plane statistics** command.

```
{primary:node0}
```

```
user@host> show chassis cluster data-plane statistics
```

```
Services Synchronized:
```

| Service name                    | RTOs sent | RTOs received |
|---------------------------------|-----------|---------------|
| Translation context             | 0         | 0             |
| Incoming NAT                    | 0         | 0             |
| Resource manager                | 6         | 0             |
| Session create                  | 161       | 0             |
| Session close                   | 148       | 0             |
| Session change                  | 0         | 0             |
| Gate create                     | 0         | 0             |
| Session ageout refresh requests | 0         | 0             |
| Session ageout refresh replies  | 0         | 0             |
| IPSec VPN                       | 0         | 0             |
| Firewall user authentication    | 0         | 0             |
| MGCP ALG                        | 0         | 0             |
| H323 ALG                        | 0         | 0             |
| SIP ALG                         | 0         | 0             |
| SCCP ALG                        | 0         | 0             |
| PPTP ALG                        | 0         | 0             |
| RPC ALG                         | 0         | 0             |
| RTSP ALG                        | 0         | 0             |
| RAS ALG                         | 0         | 0             |
| MAC address learning            | 0         | 0             |
| GPRS GTP                        | 0         | 0             |

### Verifying Chassis Cluster Redundancy Group Status

**Purpose** Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

**Action** From operational mode, enter the **chassis cluster status redundancy-group** command.

```
{primary:node0}
```

```
user@host> show chassis cluster status redundancy-group 1
```

```
Cluster ID: 1
```

| Node | Priority | Status | Preempt | Manual failover |
|------|----------|--------|---------|-----------------|
|------|----------|--------|---------|-----------------|

```
Redundancy group: 1, Failover count: 1
```

|       |     |           |    |    |
|-------|-----|-----------|----|----|
| node0 | 100 | primary   | no | no |
| node1 | 50  | secondary | no | no |

### Troubleshooting with Logs

**Purpose** Use these logs to identify any chassis cluster issues. You should run these logs on both nodes.

**Action** From operational mode, enter these **show log** commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

- Related Topics**
- Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering on page 860
  - Understanding Chassis Cluster Redundancy Groups on page 797.
  - Node Interfaces on Active SRX Series Chassis Clusters on page 846
  - Example: Configuring an SRX Series Services Gateway for the High-End as a Chassis Cluster on page 902

## Example: Configuring an SRX Series Services Gateway for the High-End as a Chassis Cluster

This example shows how to set up basic active/passive chassis clustering on a high-end SRX Series device.

- Requirements on page 902
- Overview on page 903
- Configuration on page 904
- Verification on page 912

### Requirements

Before you begin:

- You need two SRX5800 Services Gateways with identical hardware configurations, one MX240 edge router, and one EX8208 Ethernet Switch.
- Physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models. You can configure both the fabric and control ports on the SRX5000 line, for the SRX3000 line, you can configure the fabric ports only.
- Set the two devices to cluster mode and reboot the devices. You must enter the following operational mode commands on both devices, for example:
  - On node 0:

```
user@host> set chassis cluster cluster-id 1 node 0 reboot
```
  - On node 1:

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
```

The cluster-id is the same on both devices, but the node ID must be different because one device is node 0 and the other device is node 1. The range for the cluster-id is 0 through 15 and setting it to 0 is equivalent to disabling cluster mode.

If you have multiple SRX Series clusters on a single L3 broadcast domain, then you must assign different cluster IDs to each cluster, or else there will be a MAC address conflict.

From this point forward, configuration of the cluster is synchronized between the node members and the two separate devices function as one device.

### Overview

This example shows how to set up basic active/passive chassis clustering on a high-end SRX Series device. The basic active/passive example is the most common type of chassis cluster. The following high-end SRX Series devices are supported:

- SRX3400
- SRX3600
- SRX5600
- SRX5800

The basic active/passive chassis cluster consists of two devices:

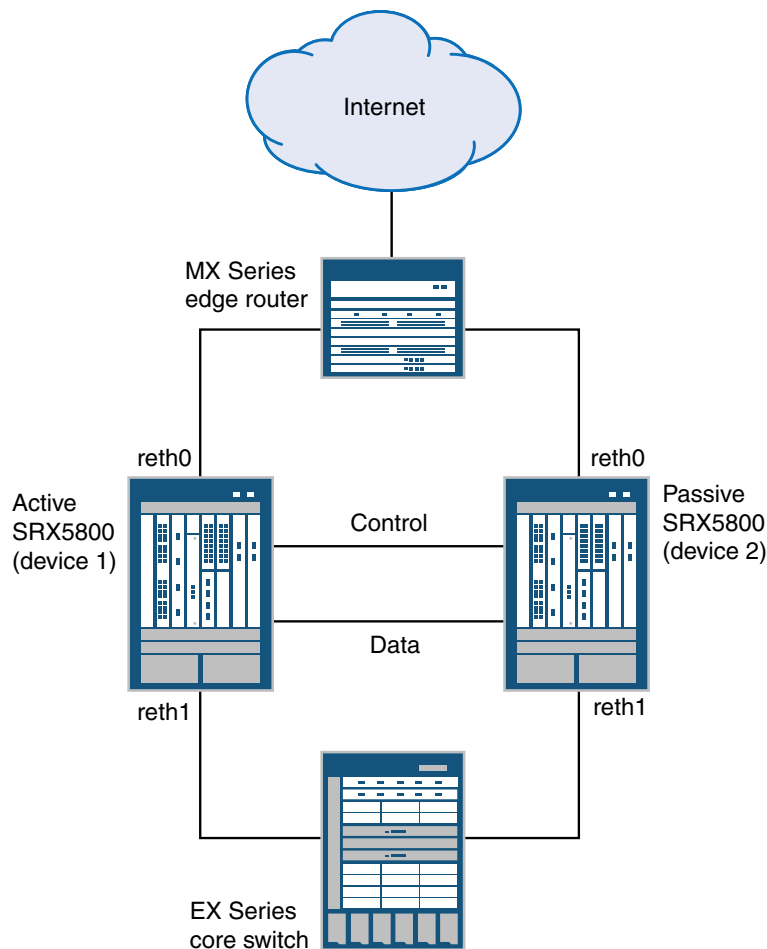
- One device actively provides routing, firewall, NAT, VPN, and security services, along with maintaining control of the chassis cluster.
- The other device passively maintains its state for cluster failover capabilities should the active device become inactive.



**NOTE:** This active/passive mode example for the SRX5800 Services Gateway does not describe in detail miscellaneous configurations such as how to configure NAT, security policies, or VPNs. They are essentially the same as they would be for standalone configurations. See “NAT Overview” on page 927, “Security Policies Overview” on page 115, and “VPN Overview” on page 355. However, if you are performing proxy ARP in chassis cluster configurations, you must apply the proxy ARP configurations to the reth interfaces rather than the member interfaces because the RETH interfaces hold the logical configurations. See “Configuring Proxy ARP (CLI Procedure)” on page 1008. You can also configure separate logical interface configurations using VLANs and trunked interfaces in the SRX5800 Services Gateway. These configurations are similar to the standalone implementations using VLANs and trunked interfaces.

Figure 106 on page 904 shows the topology used in this example.

Figure 106: Basic Active/Passive Chassis Clustering on a High-End SRX Series Device Topology Example



## Configuration

**CLI Quick Configuration** To quickly configure a chassis cluster on an SRX5800 Services Gateway, copy the following commands and paste them into the CLI:

On {primary:node0}

[edit]

```
set chassis cluster control-ports fpc 1 port 0
set chassis cluster control-ports fpc 13 port 0
set interfaces fab0 fabric-options member-interfaces ge-11/3/0
set interfaces fab1 fabric-options member-interfaces ge-23/3/0
set groups node0 system host-name SRX5800-1
set groups node0 interfaces fxp0 unit 0 family inet address 10.3.5.1/24
set groups node0 system backup-router 10.3.5.254 destination 0.0.0.0/0
set groups node1 system host-name SRX5800-2
set groups node1 interfaces fxp0 unit 0 family inet address 10.3.5.2/24
set groups node1 system backup-router 10.3.5.254 destination 0.0.0.0/0
set apply-groups "${node}"
set chassis cluster reth-count 2
```



```

set chassis cluster redundancy-group 0 node 0 priority 129
set chassis cluster redundancy-group 0 node 1 priority 128
set chassis cluster redundancy-group 1 node 0 priority 129
set chassis cluster redundancy-group 1 node 1 priority 128
set interfaces xe-6/0/0 gigether-options redundant-parent reth0
set interfaces xe-6/1/0 gigether-options redundant-parent reth1
set interfaces xe-18/0/0 gigether-options redundant-parent reth0
set interfaces xe-18/1/0 gigether-options redundant-parent reth1
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 1.1.1.1/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 2.2.2.1/24
set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-6/1/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0 weight 255
set chassis cluster control-link-recovery
set security zones security-zone untrust interfaces reth0.0
set security zones security-zone trust interfaces reth1.0
set routing-options static route 0.0.0.0/0 next-hop 1.1.1.254
set routing-options static route 2.0.0.0/8 next-hop 2.2.2.254

```

To quickly configure an EX8208 Core Switch, copy the following commands and paste them into the CLI:

On {primary:node0}

```

[edit]
set interfaces xe-1/0/0 unit 0 family ethernet-switching port-mode access vlan members
 SRX5800
set interfaces xe-2/0/0 unit 0 family ethernet-switching port-mode access vlan members
 SRX5800
set interfaces vlan unit 50 family inet address 2.2.2.254/24
set vlans SRX5800 vlan-id 50
set vlans SRX5800 l3-interface vlan.50
set routing-options static route 0.0.0.0/0 next-hop 2.2.2.1/24

```

To quickly configure an MX240 edge router, copy the following commands and paste them into the CLI:

On {primary:node0}

```

[edit]
set interfaces xe-1/0/0 encapsulation ethernet-bridge unit 0 family bridge
set interfaces xe-2/0/0 encapsulation ethernet-bridge unit 0 family bridge
set interfaces irb unit 0 family inet address 1.1.1.254/24
set routing-options static route 2.0.0.0/8 next-hop 1.1.1.1
set routing-options static route 0.0.0.0/0 next-hop (upstream router)
set bridge-domains SRX5800 vlan-id X (could be set to "none")
set bridge-domains SRX5800 domain-type bridge routing-interface irb.0
set bridge-domains SRX5800 domain-type bridge interface xe-1/0/0
set bridge-domains SRX5800 domain-type bridge interface xe-2/0/0

```

**Step-by-Step Procedure** The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a chassis cluster on a high-end SRX Series device:



**NOTE:** Perform the following steps on the primary device (node 0). They are automatically copied over to the secondary device (node 1) when you execute a **commit** command.

1. Configure control ports for the clusters. Select FPC 1/13 because the control plane should always be on the lowest SPC/SPU in the cluster (for this example, it is slot 0). For maximum reliability, place the control ports on a separate SPC from the control plane (for this example, use SPC in slot 1).



**NOTE:** For the rest of this example, all commands are applied on the control plane regardless of which member is active.

```
user@host# set chassis cluster control-ports fpc 1 port 0
user@host# set chassis cluster control-ports fpc 13 port 0
```

2. Configure the fabric (data) ports of the cluster that are used to pass RTOs in active/passive mode. For this example, use one of the 1-Gigabit Ethernet ports because running out of bandwidth using active/passive mode is not an issue. Define two fabric interfaces, one on each chassis, to connect together.

```
user@host# set interfaces fab0 fabric-options member-interfaces ge-11/3/0
user@host# set interfaces fab1 fabric-options member-interfaces ge-23/3/0
```

3. Because the SRX5800 Services Gateway chassis cluster configuration is contained within a single common configuration, to assign some elements of the configuration to a specific member only, you must use the Junos OS node-specific configuration method called groups. The **set apply-groups \${node}** command uses the node variable to define how the groups are applied to the nodes; each node recognizes its number and accepts the configuration accordingly. You must also configure out-of-band management on the fxp0 interface of the SRX5800 Services Gateway using separate IP addresses for the individual control planes of the cluster.

```
user@host# set groups node0 system host-name SRX5800-1
user@host# set groups node0 interfaces fxp0 unit 0 family inet address 10.3.5.1/24
user@host# set groups node0 system backup-router 10.3.5.254 destination 0.0.0.0/0
user@host# set groups node1 system host-name SRX5800-2
user@host# set groups node1 interfaces fxp0 unit 0 family inet address 10.3.5.2/24
user@host# set groups node1 system backup-router 10.3.5.254 destination 0.0.0.0/0
user@host# set apply-groups "${node}"
```

4. Configure redundancy groups for chassis clustering. Each node has interfaces in a redundancy group where interfaces are active in active redundancy groups (multiple active interfaces can exist in one redundancy group). Redundancy group 0 controls the control plane and redundancy group 1+ controls the data plane and includes the data plane ports. For this active/passive mode example, only one chassis cluster member is active at a time so you need to define redundancy groups 0 and 1 only. Besides redundancy groups, you must also define:
  - Redundant Ethernet groups—Configure how many redundant Ethernet interfaces (member links) will be active on the device so that the system can allocate the appropriate resources for it.
  - Priority for control plane and data plane—Define which device has priority (for chassis cluster, high priority is preferred) for the control plane, and which device is preferred to be active for the data plane.



**NOTE:** In active/passive or active/active mode, the control plane (redundancy group 0) can be active on a chassis different from the data plane (redundancy group 1+ and groups) chassis. However, for this example we recommend having both the control and data plane active on the same chassis member. When traffic passes through the fabric link to go to another member node, latency is introduced (z line mode traffic).

```
user@host# set chassis cluster reth-count 2
user@host# set chassis cluster redundancy-group 0 node 0 priority 129
user@host# set chassis cluster redundancy-group 0 node 1 priority 128
user@host# set chassis cluster redundancy-group 1 node 0 priority 129
user@host# set chassis cluster redundancy-group 1 node 1 priority 128
```

5. Configure the data interfaces on the platform so that in the event of a data plane failover, the other chassis cluster member can take over the connection seamlessly. Seamless transition to a new active node will occur with data plane failover. In case of control plane failover, all the daemons are restarted on the new node thus enabling a graceful restart to avoid losing neighborship with peers (ospf, bgp). This promotes a seamless transition to the new node without any packet loss.

You must define the following items:

- Define the membership information of the member interfaces to the reth interface.
- Define which redundancy group the reth interface is a member of. For this active/passive example, it is always 1.
- Define reth interface information such as the IP address of the interface.

```
user@host# set interfaces xe-6/0/0 gigether-options redundant-parent reth0
user@host# set interfaces xe-6/1/0 gigether-options redundant-parent reth1
user@host# set interfaces xe-18/0/0 gigether-options redundant-parent reth0
user@host# set interfaces xe-18/1/0 gigether-options redundant-parent reth1
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
user@host# set interfaces reth0 unit 0 family inet address 1.1.1.1/24
```

```

user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
user@host# set interfaces reth1 unit 0 family inet address 2.2.2.1/24

```

6. Configure the chassis cluster behavior in case of a failure. For the SRX5800 Services Gateway, the failover threshold is set at 255. You can alter the weights to determine the impact on the chassis failover. You must also configure control link recovery. The recovery automatically causes the secondary node to reboot should the control link fail, and then come back online. Enter these commands on node 0.

```

user@host# set chassis cluster redundancy-group 1 interface-monitor xe-6/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-6/1/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-18/0/0
weight 255
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-18/1/0
weight 255
user@host# set chassis cluster control-link-recovery

```

This step completes the chassis cluster configuration part of the active/passive mode example for the SRX5800 Services Gateway. The rest of this procedure describes how to configure the zone, virtual router, routing, EX8208 Core Switch, and MX240 Edge Router to complete the deployment scenario.

7. Configure and connect the reth interfaces to the appropriate zones and virtual routers. For this example, leave the reth0 and reth1 interfaces in the default virtual router inet.0, which does not require any additional configuration.

```

user@host# set security zones security-zone untrust interfaces reth0.0
user@host# set security zones security-zone trust interfaces reth1.0

```

8. For this active/passive mode example, because of the simple network architecture, use static routes to define how to route to the other network devices.

```

user@host# set routing-options static route 0.0.0.0/0 next-hop 1.1.1.254
user@host# set routing-options static route 2.0.0.0/8 next-hop 2.2.2.254

```

9. For the EX8208 Ethernet Switch, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive mode example for the SRX5800 Services Gateway; most notably the VLANs, routing, and interface configuration.

```

user@host# set interfaces xe-1/0/0 unit 0 family ethernet-switching port-mode
access vlan members SRX5800
user@host# set interfaces xe-2/0/0 unit 0 family ethernet-switching port-mode
access vlan members SRX5800
user@host# set interfaces vlan unit 50 family inet address 2.2.2.254/24
user@host# set vlans SRX5800 vlan-id 50
user@host# set vlans SRX5800 l3-interface vlan.50
user@host# set routing-options static route 0.0.0.0/0 next-hop 2.2.2.1/24

```

10. For the MX240 edge router, the following commands provide only an outline of the applicable configuration as it pertains to this active/passive mode example for the SRX5800 Services Gateway; most notably you must use an IRB interface within a virtual switch instance on the switch.

```

user@host# set interfaces xe-1/0/0 encapsulation ethernet-bridge unit 0 family
bridge
user@host# set interfaces xe-2/0/0 encapsulation ethernet-bridge unit 0 family
bridge
user@host# set interfaces irb unit 0 family inet address 1.1.1.254/24
user@host# set routing-options static route 2.0.0.0/8 next-hop 1.1.1.1
user@host# set routing-options static route 0.0.0.0/0 next-hop (upstream router)
user@host# set bridge-domains SRX5800 vlan-id X (could be set to "none")
user@host# set bridge-domains SRX5800 domain-type bridge routing-interface
irb.0
user@host# set bridge-domains SRX5800 domain-type bridge interface xe-1/0/0
user@host# set bridge-domains SRX5800 domain-type bridge interface xe-2/0/0

```

**Results** From operational mode, confirm your configuration by entering the **show configuration** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

> show configuration
version x.xx.x;
groups {
 node0 {
 system {
 host-name SRX58001;
 backup-router 10.3.5.254 destination 0.0.0.0/0;
 }
 interfaces {
 fxp0 {
 unit 0 {
 family inet {
 address 10.3.5.1/24;
 }
 }
 }
 }
 }
 node1 {
 system {
 host-name SRX58002;
 backup-router 10.3.5.254 destination 0.0.0.0/0;
 }
 interfaces {
 fxp0 {
 unit 0 {
 family inet {
 address 10.3.5.2/24;
 }
 }
 }
 }
 }
}
apply-groups "${node}";
system {
 root-authentication {
 encrypted-password "1zTMjraKG$qU8rjxoHzC6Y/WDmYpR9r.";
 }
}

```

```
name-server {
 4.2.2.2;
}
services {
 ssh {
 root-login allow;
 }
 netconf {
 ssh;
 }
 web-management {
 http {
 interface fxp0.0;
 }
 }
}
}
chassis {
 cluster {
 control-link-recovery;
 reth-count 2;
 control-ports {
 fpc 1 port 0;
 fpc 13 port 0;
 }
 redundancy-group 0 {
 node 0 priority 129;
 node 1 priority 128;
 }
 redundancy-group 1 {
 node 0 priority 129;
 node 1 priority 128;
 interface-monitor {
 xe-6/0/0 weight 255;
 xe-6/1/0 weight 255;
 xe-18/0/0 weight 255;
 xe-18/1/0 weight 255;
 }
 }
 }
}
}
interfaces {
 xe-6/0/0 {
 gigether-options {
 redundant-parent reth0;
 }
 }
 xe-6/1/0 {
 gigether-options {
 redundant-parent reth1;
 }
 }
 xe-18/0/0 {
 gigether-options {
 redundant-parent reth0;
 }
 }
 xe-18/1/0 {
 gigether-options {
 redundant-parent reth1;
 }
 }
}
```

```

}
fab0 {
 fabric-options {
 member-interfaces {
 ge-11/3/0;
 }
 }
}
fab1 {
 fabric-options {
 member-interfaces {
 ge-23/3/0;
 }
 }
}
reth0 {
 redundant-ether-options {
 redundancy-group 1;
 }
 unit 0 {
 family inet {
 address 1.1.1.1/24;
 }
 }
}
reth1 {
 redundant-ether-options {
 redundancy-group 1;
 }
 unit 0 {
 family inet {
 address 2.2.2.1/24;
 }
 }
}
}
routing-options {
 static {
 route 0.0.0.0/0 {
 next-hop 1.1.1.254;
 }
 route 2.0.0.0/8 {
 next-hop 2.2.2.254;
 }
 }
}
}
security {
 zones {
 security-zone trust {
 host-inbound-traffic {
 system-services {
 all;
 }
 }
 interfaces {
 reth0.0;
 }
 }
 security-zone untrust {
 interfaces {
 reth1.0;
 }
 }
 }
}

```

```

 }
 }
}
policies {
 from-zone trust to-zone untrust {
 policy 1 {
 match {
 source-address any;
 destination-address any;
 application any;
 }
 then {
 permit;
 }
 }
 }
}
default-policy {
 deny-all;
}
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Chassis Cluster Status on page 912
- Verifying Chassis Cluster Interfaces on page 912
- Verifying Chassis Cluster Statistics on page 913
- Verifying Chassis Cluster Control Plane Statistics on page 914
- Verifying Chassis Cluster Data Plane Statistics on page 914
- Verifying Chassis Cluster Redundancy Group Status on page 914
- Troubleshooting with Logs on page 915

#### Verifying Chassis Cluster Status

**Purpose** Verify the chassis cluster status, failover status, and redundancy group information.

**Action** From operational mode, enter the **show chassis cluster status** command.

```
{primary:node0}
show chassis cluster status
Cluster ID: 1
```

| Node                                    | Priority | Status    | Preempt | Manual failover |
|-----------------------------------------|----------|-----------|---------|-----------------|
| Redundancy group: 0 , Failover count: 1 |          |           |         |                 |
| node0                                   | 129      | primary   | no      | no              |
| node1                                   | 128      | secondary | no      | no              |
| Redundancy group: 1 , Failover count: 1 |          |           |         |                 |
| node0                                   | 129      | primary   | no      | no              |
| node1                                   | 128      | secondary | no      | no              |

#### Verifying Chassis Cluster Interfaces

**Purpose** Verify information about chassis cluster interfaces.



**Action** From operational mode, enter the **show chassis cluster interfaces** command.

```
{primary:node0}
user@host> show chassis cluster interfaces
Control link name: fxp1

Redundant-ethernet Information:
 Name Status Redundancy-group
 reth0 Up 1
 reth1 Up 1
```

```
Interface Monitoring:
 Interface Weight Status Redundancy-group
 xe-6/0/0 255 Up 1
 xe-6/1/0 255 Up 1
 xe-18/0/0 255 Up 1
 xe-18/1/0 255 Up 1
```

#### Verifying Chassis Cluster Statistics

**Purpose** Verify information about chassis cluster services and control link statistics (heartbeats sent and received), fabric link statistics (probes sent and received), and the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster statistics** command.

```
{primary:node0}
user@host> show chassis cluster statistics

Control link statistics:
 Control link 0:
 Heartbeat packets sent: 258689
 Heartbeat packets received: 258684
 Heartbeat packets errors: 0
Fabric link statistics:
 Probes sent: 258681
 Probes received: 258681
 Probe errors: 0
Services Synchronized:
 Service name RTOs sent RTOs received
 Translation context 0 0
 Incoming NAT 0 0
 Resource manager 6 0
 Session create 161 0
 Session close 148 0
 Session change 0 0
 Gate create 0 0
 Session ageout refresh requests 0 0
 Session ageout refresh replies 0 0
 IPSec VPN 0 0
 Firewall user authentication 0 0
 MGCP ALG 0 0
 H323 ALG 0 0
 SIP ALG 0 0
 SCCP ALG 0 0
 PPTP ALG 0 0
 RPC ALG 0 0
 RTSP ALG 0 0
 RAS ALG 0 0
 MAC address learning 0 0
```

|          |   |   |
|----------|---|---|
| GPRS GTP | 0 | 0 |
|----------|---|---|

### Verifying Chassis Cluster Control Plane Statistics

**Purpose** Verify information about chassis cluster control plane statistics (heartbeats sent and received) and the fabric link statistics (probes sent and received).

**Action** From operational mode, enter the **show chassis cluster control-plane statistics** command.

```
{primary:node0}
user@host> show chassis cluster control-plane statistics
```

```
Control link statistics:
 Control link 0:
 Heartbeat packets sent: 258689
 Heartbeat packets received: 258684
 Heartbeat packets errors: 0
Fabric link statistics:
 Probes sent: 258681
 Probes received: 258681
 Probe errors: 0
```

### Verifying Chassis Cluster Data Plane Statistics

**Purpose** Verify information about the number of RTOs sent and received for services.

**Action** From operational mode, enter the **show chassis cluster data-plane statistics** command.

```
{primary:node0}
user@host> show chassis cluster data-plane statistics
```

```
Services Synchronized:
 Service name RTOs sent RTOs received
 Translation context 0 0
 Incoming NAT 0 0
 Resource manager 6 0
 Session create 161 0
 Session close 148 0
 Session change 0 0
 Gate create 0 0
 Session ageout refresh requests 0 0
 Session ageout refresh replies 0 0
 IPSec VPN 0 0
 Firewall user authentication 0 0
 MGCP ALG 0 0
 H323 ALG 0 0
 SIP ALG 0 0
 SCCP ALG 0 0
 PPTP ALG 0 0
 RPC ALG 0 0
 RTSP ALG 0 0
 RAS ALG 0 0
 MAC address learning 0 0
 GPRS GTP 0 0
```

### Verifying Chassis Cluster Redundancy Group Status

**Purpose** Verify the state and priority of both nodes in a cluster and information about whether the primary node has been preempted or whether there has been a manual failover.

**Action** From operational mode, enter the **chassis cluster status redundancy-group** command.

```
{primary:node0}
user@host> show chassis cluster status redundancy-group 1
Cluster ID: 1
 Node Priority Status Preempt Manual failover

Redundancy-Group: 1, Failover count: 1
 node0 100 primary no no
 node1 50 secondary no no
```

#### Troubleshooting with Logs

**Purpose** Use these logs to identify any chassis cluster issues. You should run these logs on both nodes.

**Action** From operational mode, enter these **show log** commands.

```
user@host> show log jsrpd
user@host> show log chassisd
user@host> show log messages
user@host> show log dcd
user@host> show traceoptions
```

- Related Topics**
- Understanding Chassis Cluster Redundancy Groups on page 797.
  - Node Interfaces on Active SRX Series Chassis Clusters on page 846
  - Example: Configuring an SRX Series Services Gateway for the Branch as a Chassis Cluster on page 888

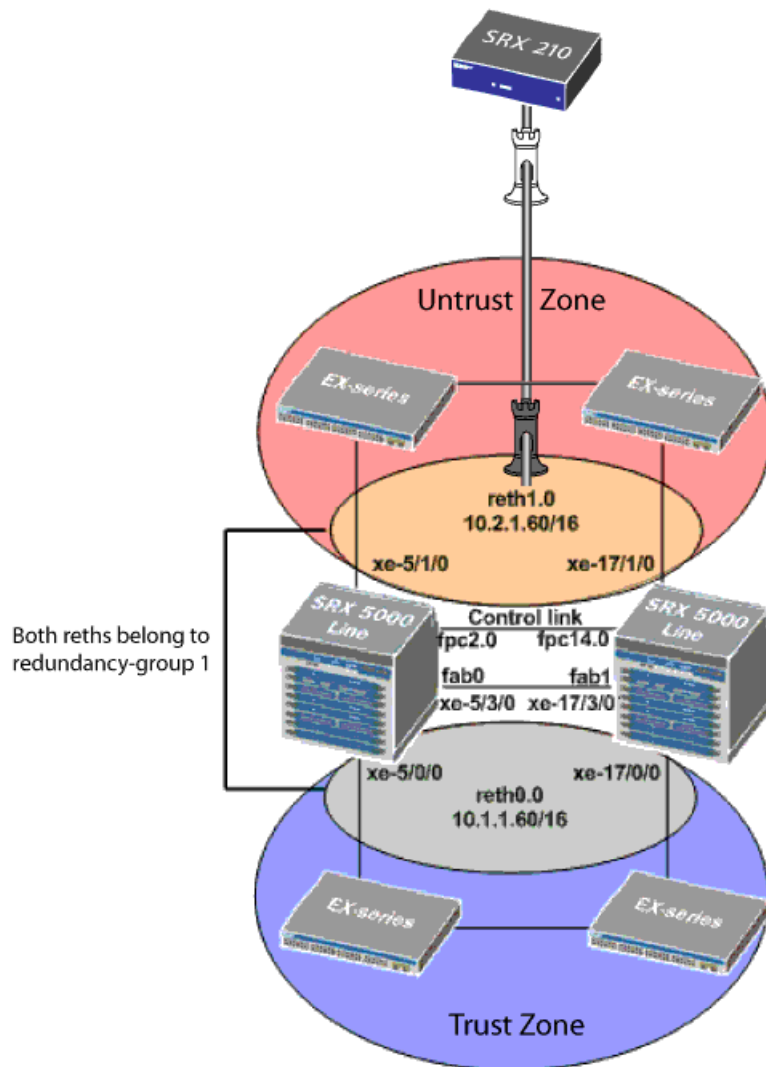
## Active/Passive Chassis Cluster Deployment with an IPsec Tunnel

- Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 915
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI) on page 917
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) on page 920

### Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel

In this case, a single device in the cluster terminates in an IPsec tunnel and is used to process all traffic while the other device is used only in the event of a failure (see Figure 107 on page 916). When a failure occurs, the backup device becomes master and controls all forwarding.

Figure 107: Active/Passive Chassis Cluster with IPsec Tunnel Scenario (SRX Series Devices)



An active/passive chassis cluster can be achieved by using redundant Ethernet interfaces (reths) that are all assigned to the same redundancy group. If any of the interfaces in an active group in a node fails, the group is declared inactive and all the interfaces in the group fail over to the other node.

This configuration provides a way for a site-to-site IPsec tunnel to terminate in an active/passive cluster where a redundant Ethernet interface is used as the tunnel endpoint. In the event of a failure, the redundant Ethernet interface in the backup SRX Series device becomes active, forcing the tunnel to change endpoints to terminate in the new active SRX Series device. Because tunnel keys and session information are synchronized between the members of the chassis cluster, a failover does not require the tunnel to be renegotiated and all established sessions are maintained.

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI) on page 917
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) on page 920
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- Understanding Chassis Cluster Formation on page 796

### Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI)

1. Enable clustering.

In node 0:

```
user@host> set chassis cluster cluster-id 1 node 0
warning: A reboot is required for chassis cluster to be enabled
```

In node 1:

These commands are applicable to SRX5600 and SRX5800 series devices.

```
user@host> set chassis cluster cluster-id 1 node 1 reboot
Successfully enabled chassis cluster. Going to reboot now.
{primary:node1}
user@host# set chassis cluster control-ports fpc 2 port 0
{primary:node1}
user@host# set chassis cluster control-ports fpc 14 port 0
```

2. Configure the management interface.

In a cluster, the configuration is shared among the cluster members. Member-specific configurations (such as the IP address of the management port of each member) are entered using configuration groups.

```
{primary:node1}
user@host# set groups node0 system host-name SRX5800-1
{primary:node1}
user@host# set groups node0 interfaces fxp0 unit 0 family inet address
172.19.100.50/24
{primary:node1}
user@host# set groups node1 system host-name SRX5800-2
{primary:node1}
user@host# set groups node1 interfaces fxp0 unit 0 family inet address
172.19.100.51/24
{primary:node1}
user@host# set apply-groups "${node}"
```

3. Configure the fabric interface.

```
{primary:node1}
user@host# set interfaces fab0 fabric-options member-interfaces xe-5/3/0
{primary:node1}
user@host# set interfaces fab1 fabric-options member-interfaces xe-17/3/0
```

4. Configure the redundancy groups.

```
{primary:node1}
```

```

user@host# set chassis cluster reth-count 2
{primary:node1}
user@host# set chassis cluster heartbeat-interval 1000
{primary:node1}
user@host# set chassis cluster heartbeat-threshold 3
{primary:node1}
user@host# set chassis cluster node 0
{primary:node1}
user@host# set chassis cluster node 1
{primary:node1}
user@host# set chassis cluster redundancy-group 0 node 0 priority 254
{primary:node1}
user@host# set chassis cluster redundancy-group 0 node 1 priority 1
{primary:node1}
user@host# set chassis cluster redundancy-group 1 node 0 priority 254
{primary:node1}
user@host# set chassis cluster redundancy-group 1 node 1 priority 1
{primary:node1}
user@host# set chassis cluster redundancy-group 1 preempt
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-5/0/0
weight 255
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-5/1/0
weight 255
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-17/0/0
weight 255
{primary:node1}
user@host# set chassis cluster redundancy-group 1 interface-monitor xe-17/1/0
weight 255

```

5. Configure the redundant Ethernet interfaces.

```

{primary:node1}
user@host# set interfaces xe-5/1/0 gigether-options redundant-parent reth1
{primary:node1}
user@host# set interfaces xe-17/1/0 gigether-options redundant-parent reth1
{primary:node1}
user@host# set interfaces xe-5/0/0 gigether-options redundant-parent reth0
{primary:node1}
user@host# set interfaces xe-17/0/0 gigether-options redundant-parent reth0
{primary:node1}
user@host# set interfaces reth0 redundant-ether-options redundancy-group 1
{primary:node1}
user@host# set interfaces reth0 unit 0 family inet address 10.1.1.60/16
{primary:node1}
user@host# set interfaces reth1 redundant-ether-options redundancy-group 1
{primary:node1}
user@host# set interfaces reth1 unit 0 family inet address 10.2.1.60/16

```

6. Configure the IPsec configuration.

```

{primary:node1}
user@host# set interfaces st0 unit 0 multipoint family inet address 10.10.1.1/30
{primary:node1}
user@host# set interfaces st0 family inet address 10.10.1.1/30

```

```

{primary:node1}
user@host# set security ike policy preShared mode main
{primary:node1}
user@host# set security ike policy preShared proposal-set standard
{primary:node1}
user@host# set security ike policy preShared pre-shared-key ascii-text "juniper"##
 Encrypted password
{primary:node1}
user@host# set security ike gateway SRX210-1 ike-policy preShared
{primary:node1}
user@host# set security ike gateway SRX210-1 address 10.1.1.90
{primary:node1}
user@host# set security ike gateway SRX210-1 external-interface reth0.0
{primary:node1}
user@host# set security ipsec policy std proposal-set standard
{primary:node1}
user@host# set security ipsec vpn SRX210-1 bind-interface st0.0
{primary:node1}
user@host# set security ipsec vpn SRX210-1 vpn-monitor optimized
{primary:node1}
user@host# set security ipsec vpn SRX210-1 ike gateway SRX210-1
{primary:node1}
user@host# set security ipsec vpn SRX210-1 ike ipsec-policy std
{primary:node1}
user@host# set security ipsec vpn SRX210-1 establish-tunnels immediately

```

7. Configure the static routes.

```

{primary:node1}
user@host# set routing-options static route 0.0.0.0/0 next-hop 10.2.1.1
{primary:node1}
user@host# set routing-options static route 10.3.0.0/16 next-hop 10.10.1.2

```

8. Configure the security zones.

```

{primary:node1}
user@host# set security zones security-zone Untrust host-inbound-traffic
 system-services all
{primary:node1}
user@host# set security zones security-zone Untrust host-inbound-traffic protocols
 all
{primary:node1}
user@host# set security zones security-zone Untrust interfaces reth1.0
{primary:node1}
user@host# set security zones security-zone Trust host-inbound-traffic
 system-services all
{primary:node1}
user@host# set security zones security-zone Trust host-inbound-traffic protocols
 all
{primary:node1}
user@host# set security zones security-zone Trust interfaces reth0.0
{primary:node1}
user@host# set security zones security-zone vpn host-inbound-traffic system-services
 all
{primary:node1}
user@host# set security zones security-zone vpn host-inbound-traffic protocols all
{primary:node1}

```

```
user@host# set security zones security-zone vpn interfaces st0.0
```

9. Configure the security policies.

```
{primary:node1}
user@host# set security policies from-zone Trust to-zone Untrust policy ANY match
source-address any
{primary:node1}
user@host# set security policies from-zone Trust to-zone Untrust policy ANY match
destination-address any
{primary:node1}
user@host# set security policies from-zone Trust to-zone Untrust policy ANY match
application any
{primary:node1}
user@host# set security policies from-zone Trust to-zone vpn policy ANY then permit
```

**Related Topics** • *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 915
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web) on page 920
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- Understanding Chassis Cluster Formation on page 796

### Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (J-Web)

1. Enable clusters. See Step 1 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI)” on page 917.
2. Configure the management interface. See Step 2 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI)” on page 917.
3. Configure the fabric interface. See Step 3 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI)” on page 917.
4. Configure the redundancy groups.
  - Select **Configure>System Properties>Chassis Cluster**.
  - Enter the following information, and then click **Apply**:  
  
Redundant ether-Interfaces Count: **2**  
  
Heartbeat Interval: **1000**  
  
Heartbeat Threshold: **3**  
  
Nodes: **0**  
  
Group Number: **0**  
  
Priorities: **254**
  - Enter the following information, and then click **Apply**:



Nodes: **0**

Group Number: **1**

Priorities: **254**

- Enter the following information, and then click **Apply**:

Nodes: **1**

Group Number: **0**

Priorities: **1**

- Enter the following information, and then click **Apply**:

Nodes: **1**

Group Number: **1**

Priorities: **1**

Preempt: Select the check box.

Interface Monitor—Interface: **xe-5/0/0**

Interface Monitor—Weight: **255**

Interface Monitor—Interface: **xe-5/1/0**

Interface Monitor—Weight: **255**

Interface Monitor—Interface: **xe-17/0/0**

Interface Monitor—Weight: **255**

Interface Monitor—Interface: **xe-17/1/0**

Interface Monitor—Weight: **255**

5. Configure the redundant Ethernet interfaces.

- Select **Configure>System Properties>Chassis Cluster**.
- Select **xe-5/1/0**.
- Enter **reth1** in the Redundant Parent box.
- Click **Apply**.
- Select **xe-17/1/0**.
- Enter **reth1** in the Redundant Parent box.
- Click **Apply**.
- Select **xe-5/0/0**.

- Enter **reth0** in the Redundant Parent box.
  - Click **Apply**.
  - Select **xe-17/0/0**.
  - Enter **reth0** in the Redundant Parent box.
  - Click **Apply**.
  - See Step 5 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI)” on page 917.
6. Configure the IPsec configuration. See Step 6 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI)” on page 917.
  7. Configure the static routes .
    - Select **Configure>Routing>Static Routing**.
    - Click **Add**.
    - Enter the following information, and then click **Apply**:  
Static Route Address: **0.0.0.0/0**  
Next-Hop Addresses: **10.2.1.1**
    - Enter the following information, and then click **Apply**:  
Static Route Address: **10.3.0.0/16**  
Next-Hop Addresses: **10.10.1.2**
  8. Configure the security zones. See Step 8 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI)” on page 917.
  9. Configure the security policies. See Step 9 in “Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI)” on page 917.
  10. Click **OK** to check your configuration and save it as a candidate configuration, then click **Commit Options>Commit**.

**Related Topics**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Active/Passive Chassis Cluster Deployment with an IPsec Tunnel on page 915
- Example: Configuring an Active/Passive Chassis Cluster Pair with an IPsec Tunnel (CLI) on page 917
- Understanding What Happens When Chassis Cluster Is Enabled on page 845
- Understanding Chassis Cluster Formation on page 796

## Limitations of Chassis Clustering

On an SRX Series or a J Series device, when defining chassis clustering, be aware of the following restrictions:

- The following features are not supported when chassis clustering is enabled on the device:
  - Packet-based forwarding for MPLS and International Organization for Standardization (ISO) protocol families.
  - Any function that depends on the configurable interfaces:
    - **lsq-0/0/0**—Link services Multilink Point-to-Point Protocol (MLPPP), Multilink Frame Relay (MLFR), and Compressed Real-Time Transport Protocol (CRTP).
    - **gr-0/0/0**—Generic routing encapsulation (GRE) and tunneling.
    - **ip-0/0/0**—IP-over-IP (IP-IP) encapsulation.
    - **pd-0/0/0**, **pe-0/0/0**, and **mt-0/0/0**—All multicast protocols.
    - **lt-0/0/0**—Real-time performance monitoring (RPM).
  - WXC Integrated Services Module (WXC ISM 200).
  - ISDN BRI
  - Layer 2 Ethernet switching

The factory default configuration for SRX100, SRX210, and SRX240 devices automatically enables Layer 2 Ethernet switching. Because Layer 2 Ethernet switching is not supported in chassis cluster mode, for these devices, if you use the factory default configuration, you must delete the Ethernet switching configuration before you enable chassis clustering.



**CAUTION:** Enabling chassis clustering while Ethernet switching is enabled is not a supported configuration. Doing so might result in undesirable behavior from the devices, leading to possible network instability.

The default configuration for other SRX Series devices and all J Series devices does not automatically enable Ethernet switching. However, if you have enabled Ethernet switching, be sure to disable it before enabling clustering on these devices too. See “Disabling Switching on SRX100, SRX210, and SRX240 Devices Before Enabling Chassis Clustering” on page 860.

- On SRX Series devices, only two of the 10 ports on each PIC of 40-port 1-Gigabit Ethernet I/O cards (IOCs) for SRX5600 and SRX5800 devices can simultaneously enable IP address monitoring. Because there are four PICs per IOC, this permits a total of eight ports per IOC to be monitored. If more than two ports per PIC on 40-port 1-Gigabit Ethernet IOCs are configured for IP address monitoring, the commit will

succeed but a log entry will be generated, and the accuracy and stability of IP address monitoring cannot be ensured. This limitation does not apply to any other IOCs or devices.

- On SRX3400, SRX3600, SRX5600, and SRX5800 devices, IP address monitoring is not permitted on redundant Ethernet interface LAGs or on child interfaces of redundant Ethernet interface LAGs.
- On SRX3000 and SRX5000 line chassis clusters, screen statistics data can be gathered on the primary device only.
- On J Series devices, a Fast Ethernet port from a 4-port Ethernet PIM cannot be used as a fabric link port in a chassis cluster.
- On SRX3400, SRX3600, SRX5600, and SRX5800 devices, in-service software upgrade (ISSU) does not support version downgrading. That is, ISSU does not support running an ISSU install of a Junos OS version that is earlier than the currently installed version.
- On SRX3400, SRX3600, SRX5600, and SRX5800 devices, only redundant Ethernet interfaces (reth) are supported for IKE external interface configuration in IPsec VPN. Other interface types can be configured but IPsec VPN might not work.

## PART 12

# Network Address Translation

- Network Address Translation on page 927



# Network Address Translation

- NAT Overview on page 927
- Understanding NAT Rule Sets and Rules on page 928
- Static NAT on page 930
- Destination NAT on page 941
- Source NAT on page 959
- Configuring Proxy ARP (CLI Procedure) on page 1008
- Verifying NAT Configuration on page 1009

## NAT Overview

---

Network Address Translation (NAT) is a method for modifying or translating network address information in packet headers. Either or both source and destination addresses in a packet may be translated. NAT can include the translation of port numbers as well as IP addresses.

NAT is described in RFC 1631 to solve IP (version 4) address depletion problems. Since then, NAT has been found to be a useful tool for firewalls, traffic redirect, load sharing, network migrations, and so on.

The following types of NAT are supported on Juniper Networks devices:

- Static NAT
- Destination NAT
- Source NAT

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding NAT Rule Sets and Rules on page 928
  - Understanding Static NAT on page 930
  - Understanding Destination NAT on page 942
  - Understanding Source NAT on page 960

## Understanding NAT Rule Sets and Rules

---

NAT processing centers on the evaluation of NAT rule sets and rules. A rule set determines the overall direction of the traffic to be processed. For example, a rule set can select traffic from a particular interface or to a specific zone. A rule set can contain multiple rules. Once a rule set is found that matches specific traffic, each rule in the rule set is evaluated for a match. Each rule in the rule set further specifies the traffic to be matched and the action to be taken when traffic matches the rule.

This topic includes the following sections:

- NAT Rule Sets on page 928
- NAT Rules on page 929
- Rule Processing on page 929

### NAT Rule Sets

A rule set specifies a general set of matching conditions for traffic. For static NAT and destination NAT, a rule set specifies one of the following:

- Source interface
- Source zone
- Source routing instance

For source NAT rule sets, you configure both source and destination conditions:

- Source interface, zone, or routing instance
- Destination interface, zone, or routing instance

It is possible for a packet to match more than one rule set; in this case, the rule set with the more specific match is used. An interface match is considered more specific than a zone match, which is more specific than a routing instance match. If a packet matches both a destination NAT rule set that specifies a source zone and a destination NAT rule set that specifies a source interface, the rule set that specifies the source interface is the more specific match.

Source NAT rule set matching is more complex because you specify both source and destination conditions in a source NAT rule set. In the case where a packet matches more than one source NAT rule set, the rule set chosen is based on the following source/destination conditions (in order of priority):

1. Source interface/destination interface
2. Source zone/destination interface
3. Source routing instance/destination interface
4. Source interface/destination zone
5. Source zone/destination zone



6. Source routing instance/destination zone
7. Source interface/destination routing instance
8. Source zone/destination routing instance
9. Source routing instance/destination routing instance

For example, you can configure rule set A, which specifies a source interface and a destination zone, and rule set B, which specifies a source zone and a destination interface. If a packet matches both rule sets, rule set B is the more specific match.



**NOTE:** You cannot specify the same source and destination conditions for source NAT rule sets.

## NAT Rules

Once a rule set that matches the traffic has been found, each rule in the rule set is evaluated in order for a match. NAT rules can match on the following packet information:

- Destination address (for static NAT only)
- Source and destination address (for destination and source NAT)
- Destination port (for destination and source NAT)

The first rule in the rule set that matches the traffic is used. If a packet matches a rule in a rule set during session establishment, traffic is processed according to the action specified by that rule.

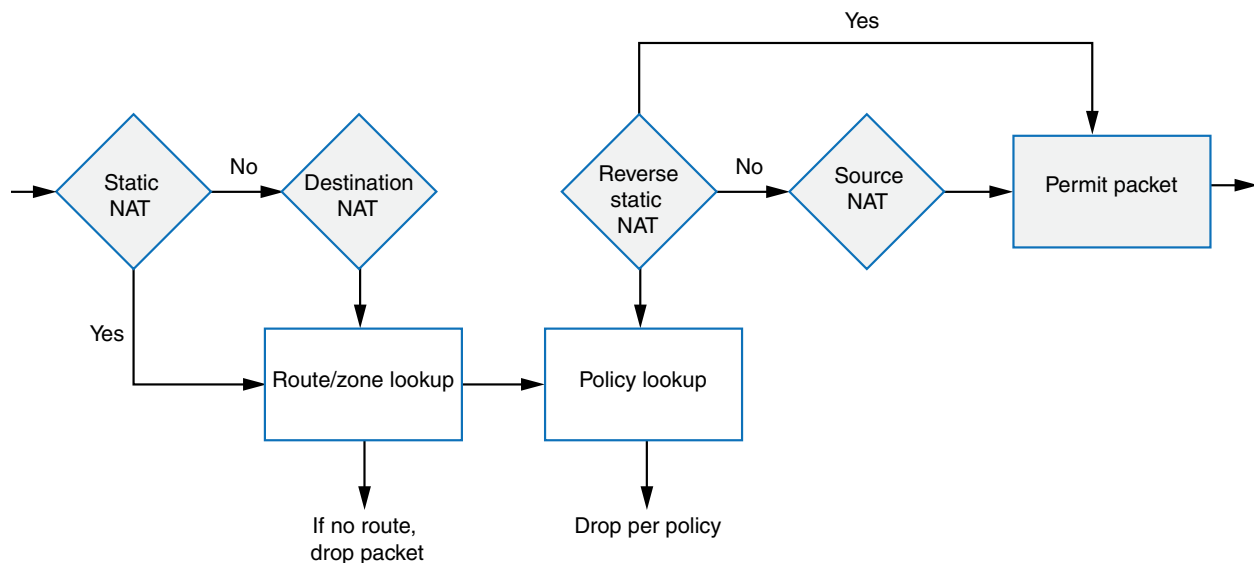
## Rule Processing

The NAT type determines the order in which NAT rules are processed. During the first packet processing for a flow, NAT rules are applied in the following order:

1. Static NAT rules
2. Destination NAT rules
3. Route lookup
4. Security policy lookup
5. Reverse mapping of static NAT rules
6. Source NAT rules

Figure 108 on page 930 illustrates the order for NAT rule processing.

Figure 108: NAT Rule Processing



Static NAT and destination NAT rules are processed before route and security policy lookup. Static NAT rules take precedence over destination NAT rules. Reverse mapping of static NAT rules takes place after route and security policy lookup and takes precedence over source NAT rules. Source NAT rules are processed after route and security policy lookup and after reverse mapping of static NAT rules.

The configuration of rules and rule sets is basically the same for each type of NAT—source, destination, or static. But because both destination and static NAT are processed before route lookup, you cannot specify the destination zone, interface or routing instance in the rule set.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - NAT Overview on page 927
  - Static NAT Configuration Overview on page 932
  - Destination NAT Configuration Overview on page 944
  - Source NAT Configuration Overview on page 965

## Static NAT

- Understanding Static NAT on page 930
- Understanding Static NAT Rules on page 931
- Static NAT Configuration Overview on page 932
- Static NAT Configuration Examples on page 932

## Understanding Static NAT

Static NAT defines a one-to-one mapping from one IP subnet to another IP subnet. The mapping includes destination IP address translation in one direction and source IP address

translation in the reverse direction. From the NAT device, the original destination address is the virtual host IP address while the mapped-to address is the real host IP address.

Static NAT allows connections to be originated from either side of the network, but translation is limited to one-to-one or between blocks of addresses of the same size. For each private address, a public address must be allocated. No address pools are necessary.



**NOTE:** The original destination address, along with other addresses in source and destination NAT pools, must not overlap within the same routing instance.

Static NAT does not perform port address translation (PAT) and no address pools are needed for static NAT.

In NAT rule lookup, static NAT rules take precedence over destination NAT rules and reverse mapping of static NAT rules take precedence over source NAT rules.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Static NAT Configuration Overview on page 932
  - Example: Configuring Static NAT for Single Address Translation on page 932
  - Example: Configuring Static NAT for Subnet Translation on page 936
  - NAT Overview on page 927
  - Understanding Static NAT Rules on page 931

## Understanding Static NAT Rules

Static NAT rules specify two layers of match conditions:

- Traffic direction—Allows you to specify **from interface**, **from zone**, or **from routing-instance**.
- Packet information—Destination IP address.

If multiple static NAT rules overlap in the match conditions, the most specific rule is chosen. For example, if rules A and B specify the same source and destination IP addresses, but rule A specifies traffic from zone 1 and rule B specifies traffic from interface **ge-0/0/0**, rule B is used to perform static NAT. An interface match is considered to be more specific than a zone match, which is more specific than a routing instance match.

For the static NAT rule action, specify the translated address and (optionally) the routing instance.

In NAT lookup, static NAT rules take precedence over destination NAT rules and reverse mapping of static NAT rules takes precedence over source NAT rules.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Static NAT on page 930
  - Static NAT Configuration Overview on page 932

- [Example: Configuring Static NAT for Single Address Translation on page 932](#)
- [Example: Configuring Static NAT for Subnet Translation on page 936](#)
- [Understanding NAT Rule Sets and Rules on page 928](#)

## Static NAT Configuration Overview

The main configuration tasks for static NAT are as follows:

1. Configure static NAT rules that align with your network and security requirements.
2. Configure NAT proxy ARP entries for IP addresses in the same subnet of the ingress interface.

### Related Topics

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Static NAT on page 930](#)
- [Configuring Proxy ARP \(CLI Procedure\) on page 1008](#)
- [Example: Configuring Static NAT for Single Address Translation on page 932](#)
- [Example: Configuring Static NAT for Subnet Translation on page 936](#)
- [Verifying NAT Configuration on page 1009](#)

## Static NAT Configuration Examples

- [Example: Configuring Static NAT for Single Address Translation on page 932](#)
- [Example: Configuring Static NAT for Subnet Translation on page 936](#)

### Example: Configuring Static NAT for Single Address Translation

This example describes how to configure a static NAT mapping of a single private address to a public address.

- [Requirements on page 932](#)
- [Overview on page 932](#)
- [Configuration on page 934](#)
- [Verification on page 936](#)

#### Requirements

Before you begin:

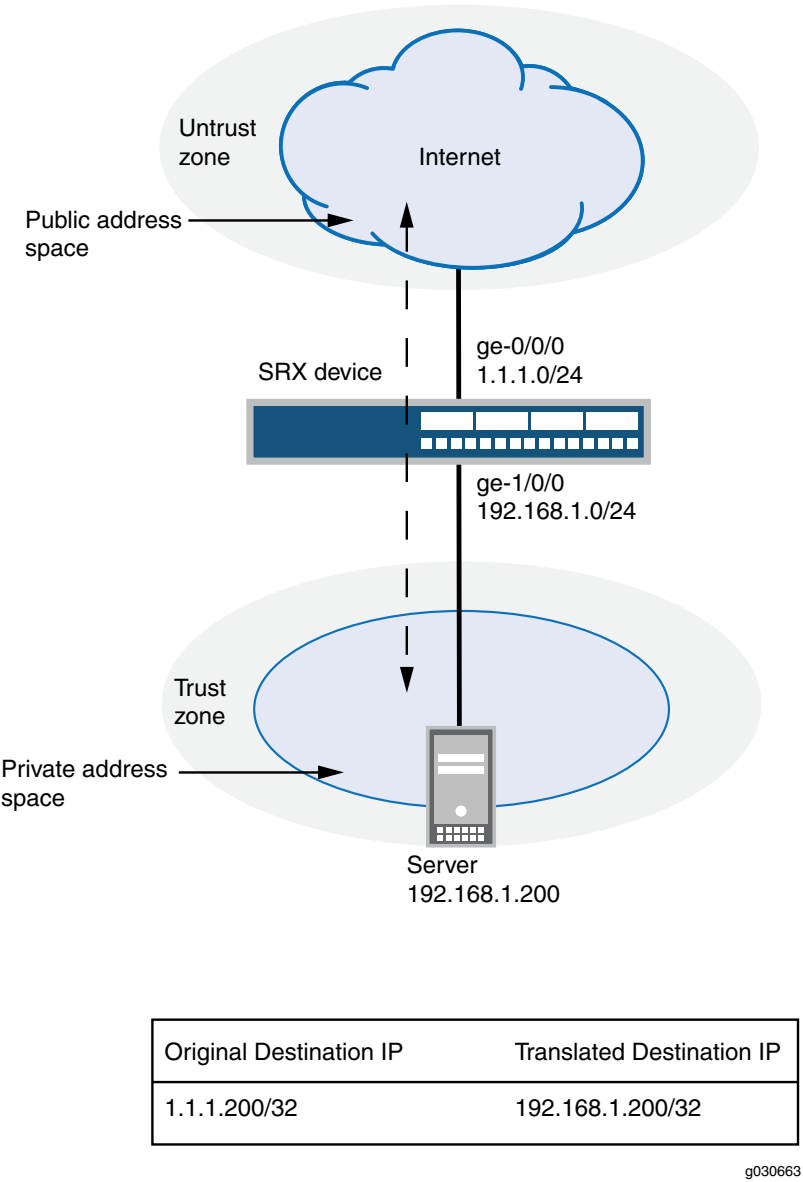
1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 87.

#### Overview

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 109 on page 933, devices in the untrust

zone access a server in the trust zone by way of public address 1.1.1.200/32. For packets that enter the Juniper Networks security device from the untrust zone with the destination IP address 1.1.1.200/32, the destination IP address is translated to the private address 192.168.1.200/32. For a new session originating from the server, the source IP address in the outgoing packet is translated to the public address 1.1.1.200/32.

Figure 109: Static NAT Single Address Translation



This example describes the following configurations:

- Static NAT rule set **rs1** with rule **r1** to match packets from the untrust zone with the destination address 1.1.1.200/32. For matching packets, the destination IP address is translated to the private address 192.168.1.200/32.

- Proxy ARP for the address 1.1.1.200 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for that address.
- Security policies to permit traffic to and from the 192.168.1.200 server.

### Configuration

**CLI Quick Configuration** To quickly configure a static NAT mapping from a private address to a public address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat static rule-set rs1 from zone untrust
set security nat static rule-set rs1 rule r1 match destination-address 1.1.1.200/32
set security nat static rule-set rs1 rule r1 then static-nat prefix 192.168.1.200/32
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
set security zones security-zone trust address-book address server-1 192.168.1.200/32
set security policies from-zone trust to-zone untrust policy permit-all match
 source-address server-1
set security policies from-zone trust to-zone untrust policy permit-all match
 destination-address any
set security policies from-zone trust to-zone untrust policy permit-all match application
 any
set security policies from-zone trust to-zone untrust policy permit-all then permit
set security policies from-zone untrust to-zone trust policy server-access match
 source-address any
set security policies from-zone untrust to-zone trust policy server-access match
 destination-address server-1
set security policies from-zone untrust to-zone trust policy server-access match application
 any
set security policies from-zone untrust to-zone trust policy server-access then permit
```

**Step-by-Step Procedure** The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a static NAT mapping from a private address to a public address:

1. Create a static NAT rule set.  

```
[edit security nat static]
user@host# set rule-set rs1 from zone untrust
```
2. Configure a rule that matches packets and translates the destination address in the packets to a private address.  

```
[edit security nat static]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.200/32
user@host# set rule-set rs1 rule r1 then static-nat prefix 192.168.1.200/32
```
3. Configure proxy ARP.  

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.200
```
4. Configure an address book entry in the trust zone for the server's IP address.

```
[edit security]
user@host# set zones security-zone trust address-book address server-1
192.168.1.200/32
```

5. Configure a security policy that allows traffic from the untrust zone to the server in the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy server-access match source-address any destination-address
server-1 application any
user@host# set policy server-access then permit
```

6. Configure a security policy that allows all traffic from the server in the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy permit-all match source-address server-1 destination-address
any application any
user@host# set policy permit-all then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
static {
 rule-set rs1 {
 from zone untrust;
 rule r1 {
 match {
 destination-address 1.1.1.200/32;
 }
 then {
 static-nat prefix 192.168.1.200/32;
 }
 }
 }
}
proxy-arp {
 interface ge-0/0/0.0 {
 address {
 1.1.1.200/32;
 }
 }
}
user@host# show security policies
from-zone trust to-zone untrust {
 policy permit-all {
 match {
 source-address server-1;
 destination-address any;
 application any;
 }
 then {
```

```
 permit;
 }
 }
 }
 from-zone untrust to-zone trust {
 policy server-access {
 match {
 source-address any;
 destination-address server-1;
 application any;
 }
 then {
 permit;
 }
 }
 }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Static NAT Configuration on page 936
- Verifying NAT Application to Traffic on page 936

### Verifying Static NAT Configuration

**Purpose** Verify that there is traffic matching the static NAT rule set.

**Action** From operational mode, enter the **show security nat static rule** command. View the Translation hits field to check for traffic that matches the rule.

### Verifying NAT Application to Traffic

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Static NAT on page 930
  - Static NAT Configuration Overview on page 932
  - Example: Configuring Static NAT for Subnet Translation on page 936

### Example: Configuring Static NAT for Subnet Translation

This example describes how to configure a static NAT mapping of a private subnet address to a public subnet address.





NOTE: Address blocks for static NAT mapping must be of the same size.

- Requirements on page 937
- Overview on page 937
- Configuration on page 939
- Verification on page 941

### Requirements

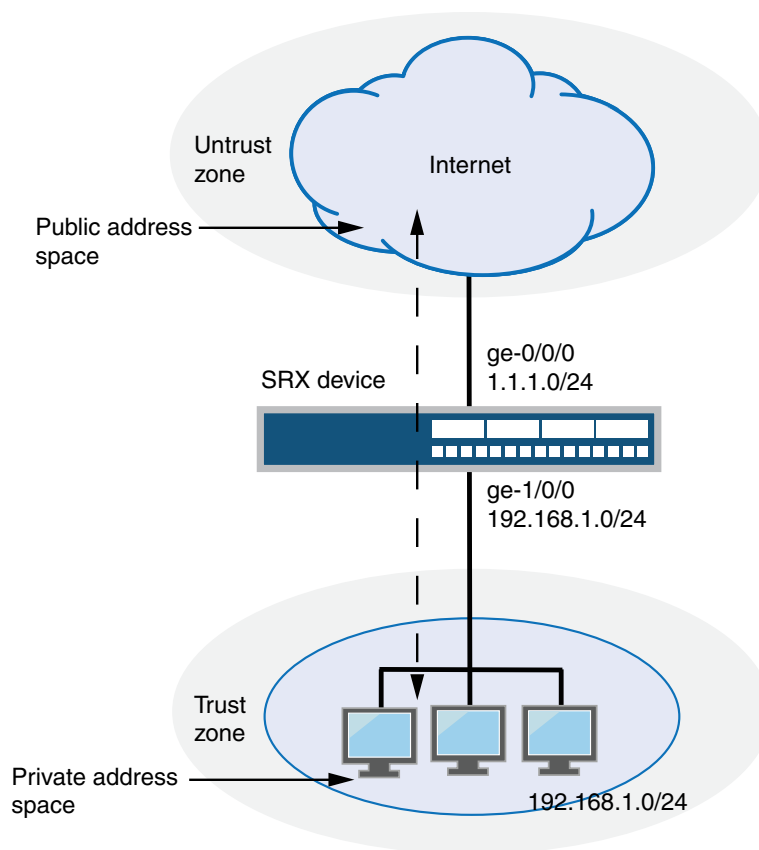
Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 87.

### Overview

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 110 on page 938, devices in the untrust zone access devices in the trust zone by way of public subnet address 1.1.1.0/24. For packets that enter the Juniper Networks security device from the untrust zone with a destination IP address in the 1.1.1.0/24 subnet, the destination IP address is translated to a private address on the 192.168.1.0/24 subnet. For new sessions originating from the 192.168.1.0/24 subnet, the source IP address in outgoing packets is translated to an address on the public 1.1.1.0/24 subnet.

Figure 110: Static NAT Subnet Translation



| Original Destination IP | Translated Destination IP |
|-------------------------|---------------------------|
| 1.1.1.0/24              | 192.168.1.0/24            |

g030664

This example describes the following configurations:

- Static NAT rule set **rs1** with rule **r1** to match packets received on interface **ge-0/0/0.0** with a destination IP address in the **1.1.1.0/24** subnet. For matching packets, the destination address is translated to an address on the **192.168.1.0/24** subnet.
- Proxy ARP for the address ranges **1.1.1.1/32** through **1.1.1.249/32** on interface **ge-0/0/0.0**. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses. The address **1.1.1.250/32** is assigned to the interface itself, so this address is not included in the proxy ARP configuration.
- Security policies to permit traffic to and from the **192.168.1.0/24** subnet.

## Configuration

**CLI Quick Configuration** To quickly configure a static NAT mapping from a private subnet address to a public subnet address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat static rule-set rs1 from interface ge-0/0/0.0
set security nat static rule-set rs1 rule r1 match destination-address 1.1.1.0/24
set security nat static rule-set rs1 rule r1 then static-nat prefix 192.168.1.0/24
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.1/32 to 1.1.1.249/32
set security zones security-zone trust address-book address server-group 192.168.1.0/24
set security policies from-zone trust to-zone untrust policy permit-all match
 source-address server-group
set security policies from-zone trust to-zone untrust policy permit-all match
 destination-address any
set security policies from-zone trust to-zone untrust policy permit-all match application
 any
set security policies from-zone trust to-zone untrust policy permit-all then permit
set security policies from-zone untrust to-zone trust policy server-access match
 source-address any
set security policies from-zone untrust to-zone trust policy server-access match
 destination-address server-group
set security policies from-zone untrust to-zone trust policy server-access match application
 any
set security policies from-zone untrust to-zone trust policy server-access then permit
```

**Step-by-Step Procedure** The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a static NAT mapping from a private subnet address to a public subnet address:

1. Create a static NAT rule set.  

```
[edit security nat static]
user@host# set rule-set rs1 from interface ge-0/0/0.0
```
2. Configure a rule that matches packets and translates the destination address in the packets to an address in a private subnet.  

```
[edit security nat static]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.0/24
user@host# set rule-set rs1 rule r1 then static-nat prefix 192.168.1.0/24
```
3. Configure proxy ARP.  

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.1/32 to 1.1.1.249/32
```
4. Configure an address book entry in the trust zone for the subnet.  

```
[edit security]
user@host# set zones security-zone trust address-book address server-group
 192.168.1.0/24
```

5. Configure a security policy that allows traffic from the untrust zone to the subnet in the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy server-access match source-address any destination-address
server-group application any
user@host# set policy server-access then permit
```

6. Configure a security policy that allows all traffic from the subnet in the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy permit-all match source-address server-group
destination-address any application any
user@host# set policy permit-all then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
static {
 rule-set rs1 {
 from interface ge-0/0/0.0;
 rule r1 {
 match {
 destination-address 1.1.0/24;
 }
 then {
 static-nat prefix 192.168.1.0/24;
 }
 }
 }
}
proxy-arp {
 interface ge-0/0/0.0 {
 address {
 1.1.1/32;
 }
 }
}
user@host# show security policies
from-zone trust to-zone untrust {
 policy permit-all {
 match {
 source-address server-group;
 destination-address any;
 application any;
 }
 then {
 permit;
 }
 }
}
```

```

}
from-zone untrust to-zone trust {
 policy server-access {
 match {
 source-address any;
 destination-address server-group;
 application any;
 }
 then {
 permit;
 }
 }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

### Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Static NAT Configuration on page 941
- Verifying NAT Application to Traffic on page 941

### Verifying Static NAT Configuration

**Purpose** Verify that there is traffic matching the static NAT rule set.

**Action** From operational mode, enter the **show security nat static rule** command. View the Translation hits field to check for traffic that matches the rule.

### Verifying NAT Application to Traffic

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Static NAT on page 930
  - Static NAT Configuration Overview on page 932
  - Example: Configuring Static NAT for Single Address Translation on page 932

## Destination NAT

- Understanding Destination NAT on page 942
- Understanding Destination NAT Address Pools on page 942
- Understanding Destination NAT Rules on page 943
- Destination NAT Configuration Overview on page 944
- Destination NAT Configuration Examples on page 944

## Understanding Destination NAT

Destination NAT is the translation of the destination IP address of a packet entering the Juniper Networks device. Destination NAT is used to redirect traffic destined to a virtual host (identified by the original destination IP address) to the real host (identified by the translated destination IP address).



---

**NOTE:** When destination NAT is performed, the destination IP address is translated according to configured destination NAT rules and then security policies are applied.

---

Destination NAT allows connections to be initiated only for incoming network connections—for example, from the Internet to a private network. Destination NAT is commonly used to perform the following actions:

- Translate a single IP address to another address (for example, to allow a device on the Internet to connect to a host on a private network).
- Translate a contiguous block of addresses to another block of addresses of the same size (for example, to allow access to a group of servers).
- Translate a destination IP address and port to another destination IP address and port (for example, to allow access to multiple services using the same IP address but different ports).

The following types of destination NAT are supported:

- Translation of the original destination IP address to an IP address from a user-defined pool. This type of translation does not include Port Address Translation (PAT). If the original destination IP address range is larger than the address range in the user-defined address pool, any untranslated packets are dropped.
- Translation of the original destination IP address (and optional port number) to one specific IP address (and port number) from a user-defined pool.

### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Destination NAT Configuration Overview on page 944
- Example: Configuring Destination NAT for Single Address Translation on page 944
- Example: Configuring Destination NAT for IP Address and Port Translation on page 949
- Example: Configuring Destination NAT for Subnet Translation on page 955
- NAT Overview on page 927
- Understanding Destination NAT Address Pools on page 942
- Understanding Destination NAT Rules on page 943

## Understanding Destination NAT Address Pools

For destination NAT address pools, specify the following:

- Name of the destination NAT address pool
- Destination address or address range



**NOTE:** Do not overlap NAT addresses for source NAT, destination NAT, and static NAT within one routing instance.

- Destination port that is used for port forwarding
- Routing instance to which the pool belongs (the default is the main **inet.0** routing instance)

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Destination NAT on page 942
- Destination NAT Configuration Overview on page 944
- Example: Configuring Destination NAT for Single Address Translation on page 944
- Example: Configuring Destination NAT for IP Address and Port Translation on page 949
- Example: Configuring Destination NAT for Subnet Translation on page 955

## Understanding Destination NAT Rules

Destination NAT rules specify two layers of match conditions:

- Traffic direction—Allows you to specify **from interface**, **from zone**, or **from routing-instance**.
- Packet information—Can be source IP addresses, destination IP address or subnet, or a single destination port number.

If multiple destination NAT rules overlap in the match conditions, the most specific rule is chosen. For example, if rules A and B specify the same source and destination IP addresses, but rule A specifies traffic from zone 1 and rule B specifies traffic from interface **ge-0/0/0**, rule B is used to perform destination NAT. An interface match is considered to be more specific than a zone match, which is more specific than a routing instance match.

The actions you can specify for a destination NAT rule are:

- **off**—Do not perform destination NAT.
- **pool**—Use the specified user-defined address pool to perform destination NAT.

Destination NAT rules are applied to traffic in the first packet that is processed for the flow or in the fast path for the ALG. Destination NAT rules are processed after static NAT rules but before source NAT rules.

#### Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Destination NAT on page 942

- [Destination NAT Configuration Overview on page 944](#)
- [Example: Configuring Destination NAT for Single Address Translation on page 944](#)
- [Example: Configuring Destination NAT for IP Address and Port Translation on page 949](#)
- [Example: Configuring Destination NAT for Subnet Translation on page 955](#)
- [Understanding NAT Rule Sets and Rules on page 928](#)

## Destination NAT Configuration Overview

The main configuration tasks for destination NAT are as follows:

1. Configure a destination NAT address pool that aligns with your network and security requirements.
2. Configure destination NAT rules that align with your network and security requirements.
3. Configure NAT proxy ARP entries for IP addresses in the same subnet of the ingress interface.

### Related Topics

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [Understanding Destination NAT on page 942](#)
- [Configuring Proxy ARP \(CLI Procedure\) on page 1008](#)
- [Example: Configuring Destination NAT for Single Address Translation on page 944](#)
- [Example: Configuring Destination NAT for IP Address and Port Translation on page 949](#)
- [Example: Configuring Destination NAT for Subnet Translation on page 955](#)
- [Verifying NAT Configuration on page 1009](#)

## Destination NAT Configuration Examples

- [Example: Configuring Destination NAT for Single Address Translation on page 944](#)
- [Example: Configuring Destination NAT for IP Address and Port Translation on page 949](#)
- [Example: Configuring Destination NAT for Subnet Translation on page 955](#)

### Example: Configuring Destination NAT for Single Address Translation

This example describes how to configure a destination NAT mapping of a single public address to a private address.





NOTE: Mapping one destination IP address to another can also be accomplished with static NAT. Static NAT mapping allows connections to be established from either side of the gateway device, whereas destination NAT only allows connections to be established from one side. However, static NAT only allows translations from one address to another or between blocks of addresses of the same size.

- Requirements on page 945
- Overview on page 945
- Configuration on page 947
- Verification on page 949

### Requirements

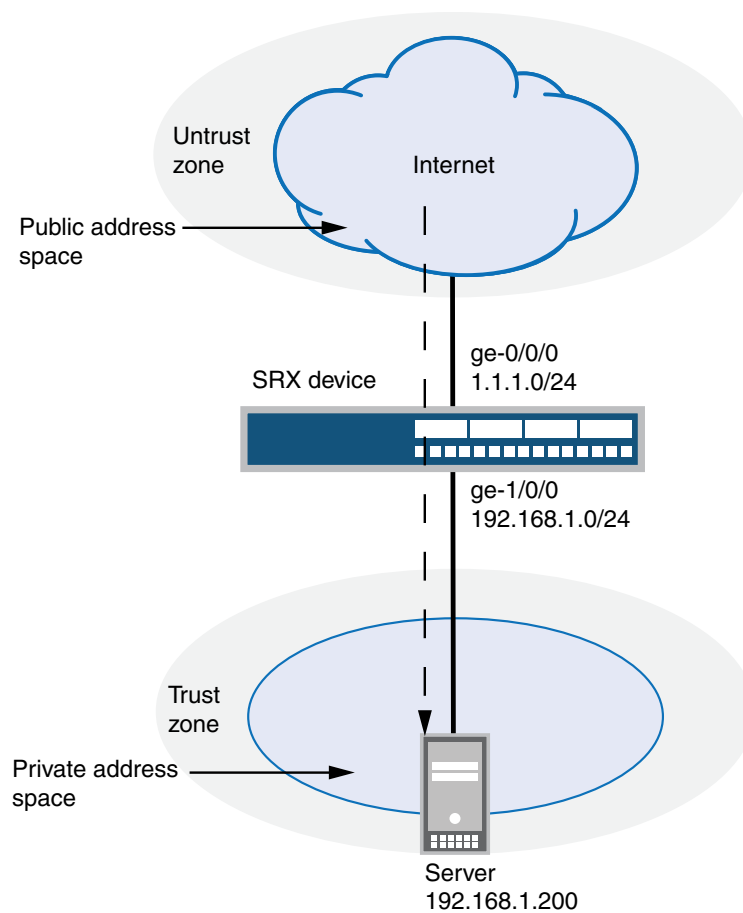
Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 87.

### Overview

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 111 on page 946, devices in the untrust zone access a server in the trust zone by way of public address 1.1.1.200/32. For packets that enter the Juniper Networks security device from the untrust zone with the destination IP address 1.1.1.200/32, the destination IP address is translated to the private address 192.168.1.200/32.

Figure 111: Destination NAT Single Address Translation



| Original Destination IP | Translated Destination IP |
|-------------------------|---------------------------|
| 1.1.1.200/32            | 192.168.1.200/32          |

g030665

This example describes the following configurations:

- Destination NAT pool **dst-nat-pool-1** that contains the IP address 192.168.1.200/32.
- Destination NAT rule set **rs1** with rule **r1** to match packets received from the ge-0/0/0.0 interface with the destination IP address 1.1.1.200/32. For matching packets, the destination address is translated to the address in the **dst-nat-pool-1** pool.
- Proxy ARP for the address 1.1.1.200/32 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for that address.
- Security policies to permit traffic from the untrust zone to the translated destination IP address in the trust zone.

**Configuration****CLI Quick  
Configuration**

To quickly configure a destination NAT mapping from a public address to a private address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat destination pool dst-nat-pool-1 address 192.168.1.200/32
set security nat destination rule-set rs1 from interface ge-0/0/0.0
set security nat destination rule-set rs1 rule r1 match destination-address 1.1.1.200/32
set security nat destination rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
set security zones security-zone trust address-book address server-1 192.168.1.200/32
set security policies from-zone untrust to-zone trust policy server-access match
 source-address any
set security policies from-zone untrust to-zone trust policy server-access match
 destination-address server-1
set security policies from-zone untrust to-zone trust policy server-access match application
 any
set security policies from-zone untrust to-zone trust policy server-access then permit
```

**Step-by-Step  
Procedure**

The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a destination NAT mapping from a public address to a private address:

1. Create the destination NAT pool.  

```
[edit security nat destination]
user@host# set pool dst-nat-pool-1 address 192.168.1.200/32
```
2. Create a destination NAT rule set.  

```
[edit security nat destination]
user@host# set rule-set rs1 from interface ge-0/0/0.0
```
3. Configure a rule that matches packets and translates the destination address to the address in the pool.  

```
[edit security nat destination]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.200/32
user@host# set rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
```
4. Configure proxy ARP.  

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
```
5. Configure an address book entry in the trust zone for the server.  

```
[edit security]
user@host# set zones security-zone trust address-book address server-1
192.168.1.200/32
```
6. Configure a security policy that allows traffic from the untrust zone to the server in the trust zone.  

```
[edit security policies from-zone untrust to-zone trust]
```

```
user@host# set policy server-access match source-address any destination-address
server-1 application any
user@host# set policy server-access then permit
```

**Results** From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
destination {
 pool dst-nat-pool-1 {
 address 192.168.1.200/32;
 }
 rule-set rs1 {
 from interface ge-0/0/0.0;
 rule r1 {
 match {
 destination-address 1.1.1.200/32;
 }
 then {
 destination-nat pool dst-nat-pool-1;
 }
 }
 }
}
}
proxy-arp {
 interface ge-0/0/0.0 {
 address {
 1.1.1.200/32;
 }
 }
}
user@host# show security policies
from-zone untrust to-zone trust {
 policy server-access {
 match {
 source-address any;
 destination-address server-1;
 application any;
 }
 then {
 permit;
 }
 }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

**Verification**

To confirm that the configuration is working properly, perform these tasks:

- Verifying Destination NAT Pool Usage on page 949
- Verifying Destination NAT Rule Usage on page 949
- Verifying NAT Application to Traffic on page 949

**Verifying Destination NAT Pool Usage**

**Purpose** Verify that there is traffic using IP addresses from the destination NAT pool.

**Action** From operational mode, enter the **show security nat destination pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

**Verifying Destination NAT Rule Usage**

**Purpose** Verify that there is traffic matching the destination NAT rule.

**Action** From operational mode, enter the **show security nat destination rule all** command. View the Translation hits field to check for traffic that matches the rule.

**Verifying NAT Application to Traffic**

**Purpose** Verify that NAT is being applied to the specified traffic.

**Action** From operational mode, enter the **show security flow session** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
  - Understanding Destination NAT on page 942
  - Destination NAT Configuration Overview on page 944
  - Example: Configuring Destination NAT for IP Address and Port Translation on page 949
  - Example: Configuring Destination NAT for Subnet Translation on page 955

**Example: Configuring Destination NAT for IP Address and Port Translation**

This example describes how to configure destination NAT mappings of a public address to private addresses, depending on the port number.

- Requirements on page 949
- Overview on page 950
- Configuration on page 952
- Verification on page 954

**Requirements**

Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.

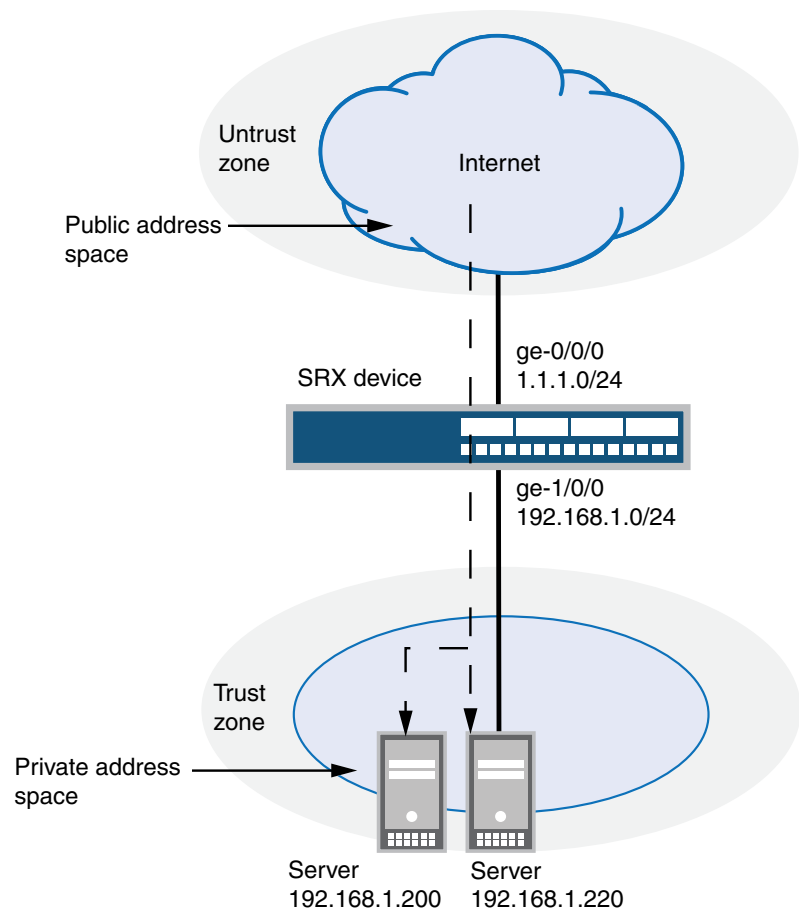
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 87.

### **Overview**

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 112 on page 951, devices in the untrust zone access servers in the trust zone by way of public address 1.1.1.200 on port 80 or 8000. Packets entering the Juniper Networks security device from the untrust zone are mapped to the private addresses of the servers as follows:

- The destination IP address 1.1.1.200 and port 80 is translated to the private address 192.168.1.200 and port 80.
- The destination IP address 1.1.1.200 and port 8000 is translated to the private address 192.168.1.220 and port 8000.

Figure 112: Destination NAT Address and Port Translation



| Original Destination IP | Translated Destination IP  |
|-------------------------|----------------------------|
| 1.1.1.200<br>port 80    | 192.168.1.200<br>port 80   |
| 1.1.1.200<br>port 8000  | 192.168.1.220<br>port 8000 |

g030666

This example describes the following configurations:

- Destination NAT pool **dst-nat-pool-1** that contains the IP address 192.168.1.200 port 80.
- Destination NAT pool **dst-nat-pool-2** that contains the IP address 192.168.1.220 and port 8000.
- Destination NAT rule set **rs1** with rule **r1** to match packets received from the untrust zone with the destination IP address 1.1.1.200 and destination port 80. For matching packets, the destination address is translated to the address in the **dst-nat-pool-1** pool.

- Destination NAT rule set **rs1** with rule **r2** to match packets received from the untrust zone with the destination IP address 1.1.1.200 and destination port 8000. For matching packets, the destination IP address and port are translated to the address and port in the **dst-nat-pool-2** pool.
- Proxy ARP for the address 1.1.1.200/32. This allows the Juniper Networks security device to respond to ARP requests received on the interface for that address.
- Security policies to permit traffic from the untrust zone to the translated destination IP addresses in the trust zone.

### Configuration

#### CLI Quick Configuration

To quickly configure a destination NAT mapping from a public address to a private address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat destination pool dst-nat-pool-1 address 192.168.1.200/32
set security nat destination pool dst-nat-pool-1 address port 80
set security nat destination pool dst-nat-pool-2 address 192.168.1.220/32
set security nat destination pool dst-nat-pool-2 address port 8000
set security nat destination rule-set rs1 from zone untrust
set security nat destination rule-set rs1 rule r1 match destination-address 1.1.1.200/32
set security nat destination rule-set rs1 rule r1 match destination-port 80
set security nat destination rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
set security nat destination rule-set rs1 rule r2 match destination-address 1.1.1.200/32
set security nat destination rule-set rs1 rule r2 match destination-port 8000
set security nat destination rule-set rs1 rule r2 then destination-nat pool dst-nat-pool-2
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
set security zones security-zone trust address-book address server-2 192.168.1.220/32
set security zones security-zone tru address-book address server-1 192.168.1.200/32
set security policies from-zone untrust to-zone trust policy server-access match
 source-address any
set security policies from-zone untrust to-zone trust policy server-access match
 destination-address server-1
set security policies from-zone untrust to-zone trust policy server-access match
 destination-address server-2
set security policies from-zone untrust to-zone trust policy server-access match application
 any
set security policies from-zone untrust to-zone trust policy server-access then permit
```

#### Step-by-Step Procedure

The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a destination NAT mapping from a public address to a private address:

1. Create destination NAT pools.

```
[edit security nat destination]
user@host# set pool dst-nat-pool-1 address 192.168.1.200 port 80
user@host# set pool dst-nat-pool-2 address 192.168.1.220 port 8000
```

2. Create a destination NAT rule set.



- ```
[edit security nat destination]
user@host# set rule-set rs1 from zone untrust
```
3. Configure a rule that matches packets and translates the destination address to the address in the pool.


```
[edit security nat destination]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.200
user@host# set rule-set rs1 rule r1 match destination-port 80
user@host# set rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
```
 4. Configure a rule that matches packets and translates the destination address to the address in the pool.


```
[edit security nat destination]
user@host# set rule-set rs1 rule r2 match destination-address 1.1.1.200
user@host# set rule-set rs1 rule r2 match destination-port 8000
user@host# set rule-set rs1 rule r2 then destination-nat pool dst-nat-pool-2
```
 5. Configure proxy ARP.


```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
```
 6. Configure address book entries in the trust zone for the server addresses.


```
[edit security]
user@host# set zones security-zone trust address-book address server-1
192.168.1.200/32
user@host# set zones security-zone trust address-book address server-2
192.168.1.220/32
```
 7. Configure a security policy that allows traffic from the untrust zone to the servers in the trust zone.


```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy server-access match source-address any destination-address
[server-1 server-2] application any
user@host# set policy server-access then permit
```

Results From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
destination {
  pool dst-nat-pool-1 {
    address 192.168.1.200/32 port 80;
  }
  pool dst-nat-pool-2 {
    address 192.168.1.220/32 port 8000;
  }
}
rule-set rs1 {
  from zone untrust;
  rule r1 {
    match {
```

```

        destination-address 1.1.1.200/32;
        destination-port 80;
    }
    then {
        destination-nat pool dst-nat-pool-1;
    }
}
rule r2 {
    match {
        destination-address 1.1.1.200/32;
        destination-port 8000;
    }
    then {
        destination-nat pool dst-nat-pool-2;
    }
}
}
}
}
proxy-arp {
    interface ge-0/0/0.0 {
        address {
            1.1.1.200/32;
        }
    }
}
}
user@host# show security policies
from-zone untrust to-zone trust {
    policy server-access {
        match {
            source-address any;
            destination-address [ server-1 server-2 ];
            application any;
        }
        then {
            permit;
        }
    }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Destination NAT Pool Usage on page 954
- Verifying Destination NAT Rule Usage on page 955
- Verifying NAT Application to Traffic on page 955

Verifying Destination NAT Pool Usage

Purpose Verify that there is traffic using IP addresses from the destination NAT pool.

Action From operational mode, enter the **show security nat destination pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

Verifying Destination NAT Rule Usage

Purpose Verify that there is traffic matching the destination NAT rule.

Action From operational mode, enter the **show security nat destination rule all** command. View the Translation hits field to check for traffic that matches the rule.

Verifying NAT Application to Traffic

Purpose Verify that NAT is being applied to the specified traffic.

Action From operational mode, enter the **show security flow session** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Destination NAT on page 942
 - Destination NAT Configuration Overview on page 944
 - Example: Configuring Destination NAT for Single Address Translation on page 944
 - Example: Configuring Destination NAT for Subnet Translation on page 955

Example: Configuring Destination NAT for Subnet Translation

This example describes how to configure a destination NAT mapping of a public subnet address to a private subnet address.



NOTE: Mapping addresses from one subnet to another can also be accomplished with static NAT. Static NAT mapping allows connections to be established from either side of the gateway device, whereas destination NAT allows connections to be established from only one side. However, static NAT only allows translations between blocks of addresses of the same size.

- Requirements on page 955
- Overview on page 955
- Configuration on page 957
- Verification on page 959

Requirements

Before you begin:

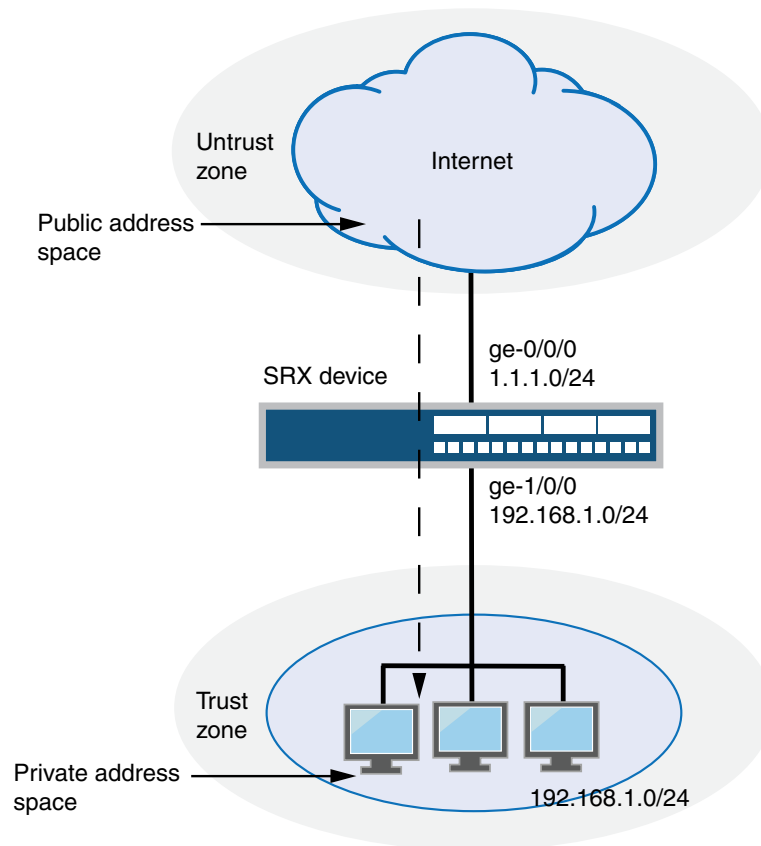
1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 87.

Overview

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 113 on page 956, devices in the untrust

zone access devices in the trust zone by way of public subnet address 1.1.1.0/16. For packets that enter the Juniper Networks security device from the untrust zone with a destination IP address in the 1.1.1.0/16 subnet, the destination IP address is translated to a private address on the 192.168.1.0/24 subnet.

Figure 113: Destination NAT Subnet Translation



Original Destination IP	Translated Destination IP
1.1.1.0/16	192.168.1.0/24

g030667

This example describes the following configurations:

- Destination NAT pool **dst-nat-pool-1** that contains the IP address 192.168.1.0/24.
- Destination NAT rule set **rs1** with rule **r1** to match packets received from the ge-0/0/0.0 interface with the destination IP address on the 1.1.1.0/16 subnet. For matching packets, the destination address is translated to the address in the **dst-nat-pool-1** pool.
- Proxy ARP for the addresses 1.1.1.1/32 through 1.1.1.62/32 on the interface ge-0/0/0.0; these are the IP addresses of the hosts that should be translated from the 1.1.1.0/16

subnet. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses. The address 1.1.1.63/32 is assigned to the interface itself, so this address is not included in the proxy ARP configuration. The addresses that are not in the 1.1.1.1/32 through 1.1.1.62/32 range are not expected to be present on the network and would not be translated.

- Security policies to permit traffic from the untrust zone to the translated destination IP addresses in the trust zone.

Configuration

CLI Quick Configuration To quickly configure a destination NAT mapping from a public subnet address to a private subnet address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat destination pool dst-nat-pool-1 address 192.168.1.0/24
set security nat destination rule-set rs1 from interface ge-0/0/0.0
set security nat destination rule-set rs1 rule r1 match destination-address 1.1.1.0/16
set security nat destination rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.1/32 to 1.1.1.62/32
set security zones security-zone trust address-book address internal-net 192.168.1.0/24
set security policies from-zone untrust to-zone trust policy internal-access match
  source-address any
set security policies from-zone untrust to-zone trust policy internal-access match
  destination-address internal-net
set security policies from-zone untrust to-zone trust policy internal-access match
  application any
set security policies from-zone untrust to-zone trust policy internal-access then permit
```

Step-by-Step Procedure The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode.

To configure a destination NAT mapping from a public subnet address to a private subnet address:

1. Create the destination NAT pool.

```
[edit security nat destination]
user@host# set pool dst-nat-pool-1 address 192.168.1.0/24
```
2. Create a destination NAT rule set.

```
[edit security nat destination]
user@host# set rule-set rs1 from interface ge-0/0/0.0
```
3. Configure a rule that matches packets and translates the destination address to an address in the pool.

```
[edit security nat destination]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.0/16
user@host# set rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
```
4. Configure proxy ARP.

```
[edit security nat]
```

```
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.1/32 to 1.1.1.62/32
```

5. Configure an address book entry in the trust zone for the private subnet address.

```
[edit security]
```

```
user@host# set zones security-zone trust address-book address internal-net  
192.168.1.0/24
```

6. Configure a security policy that allows traffic from the untrust zone to the devices in the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
```

```
user@host# set policy internal-access match source-address any  
destination-address internal-net application any  
user@host# set policy internal-access then permit
```

Results From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
```

```
user@host# show security nat
```

```
destination {  
  pool dst-nat-pool-1 {  
    address 192.168.1.0/24;  
  }  
  rule-set rs1 {  
    from interface ge-0/0/0.0;  
    rule r1 {  
      match {  
        destination-address 1.1.1.0/16;  
      }  
      then {  
        destination-nat pool dst-nat-pool-1;  
      }  
    }  
  }  
}  
proxy-arp {  
  interface ge-0/0/0.0 {  
    address {  
      1.1.1.1/32 to 1.1.1.62/32;  
    }  
  }  
}
```

```
user@host# show security policies
```

```
from-zone untrust to-zone trust {  
  policy internal-access {  
    match {  
      source-address any;  
      destination-address internal-net;  
      application any;  
    }  
    then {  
      permit;  
    }  
  }  
}
```

```

    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Destination NAT Pool Usage on page 959
- Verifying Destination NAT Rule Usage on page 959
- Verifying NAT Application to Traffic on page 959

Verifying Destination NAT Pool Usage

Purpose Verify that there is traffic using IP addresses from the destination NAT pool.

Action From operational mode, enter the **show security nat destination pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

Verifying Destination NAT Rule Usage

Purpose Verify that there is traffic matching the destination NAT rule.

Action From operational mode, enter the **show security nat destination rule all** command. View the Translation hits field to check for traffic that matches the rule.

Verifying NAT Application to Traffic

Purpose Verify that NAT is being applied to the specified traffic.

Action From operational mode, enter the **show security flow session** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Destination NAT on page 942
 - Destination NAT Configuration Overview on page 944
 - Example: Configuring Destination NAT for Single Address Translation on page 944
 - Example: Configuring Destination NAT for IP Address and Port Translation on page 949

Source NAT

- Understanding Source NAT on page 960
- Source NAT Pools on page 961
- Understanding Source NAT Rules on page 964
- Source NAT Configuration Overview on page 965
- Source NAT Configuration Examples on page 965

- Disabling Port Randomization for Source NAT (CLI Procedure) on page 1002
- Persistent NAT on page 1003

Understanding Source NAT

Source NAT is the translation of the source IP address of a packet leaving the Juniper Networks device. Source NAT is used to allow hosts with private IP addresses to access a public network.

Source NAT allows connections to be initiated only for outgoing network connections—for example, from a private network to the Internet. Source NAT is commonly used to perform the following translations:

- Translate a single IP address to another address (for example, to provide a single device in a private network with access to the Internet).
- Translate a contiguous block of addresses to another block of addresses of the same size.
- Translate a contiguous block of addresses to another block of addresses of smaller size.
- Translate a contiguous block of addresses to a single IP address or a smaller block of addresses using port translation.
- Translate a contiguous block of addresses to the address of the egress interface.

Translation to the address of the egress interface does not require an address pool; all other source NAT translations require configuration of an address pool. One-to-one and many-to-many translations for address blocks of the same size do not require port translation because there is an available address in the pool for every address that would be translated.

If the size of the address pool is smaller than the number of addresses that would be translated, either the total number of concurrent addresses that can be translated is limited by the size of the address pool or port translation must be used. For example, if a block of 253 addresses is translated to an address pool of 10 addresses, a maximum of 10 devices can be connected concurrently unless port translation is used.

The following types of source NAT are supported:

- Translation of the original source IP address to the egress interface's IP address (also called interface NAT). Port address translation is always performed.
- Translation of the original source IP address to an IP address from a user-defined address pool without port address translation. The association between the original source IP address to the translated source IP address is dynamic. However, once there is an association, the same association is used for the same original source IP address for new traffic that matches the same NAT rule.
- Translation of the original source IP address to an IP address from a user-defined address pool with port address translation. The association between the original source IP address to the translated source IP address is dynamic. Even if an association exists,

the same original source IP address may be translated to a different address for new traffic that matches the same NAT rule.

- Translation of the original source IP address to an IP address from a user-defined address pool by shifting the IP addresses. This type of translation is one-to-one, static, and without port address translation. If the original source IP address range is larger than the IP address range in the user-defined pool, untranslated packets are dropped.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Source NAT Pools on page 961
 - Understanding Source NAT Rules on page 964
 - Source NAT Configuration Overview on page 965
 - NAT Overview on page 927

Source NAT Pools

- Understanding Source NAT Pools on page 961
- Understanding Source NAT Pools with PAT on page 962
- Understanding Source NAT Pools Without PAT on page 963
- Understanding Source NAT Pools with Address Shifting on page 963
- Understanding Persistent Addresses on page 964

Understanding Source NAT Pools

For source NAT address pools, specify the following:

- Name of the source NAT address pool.
- Up to eight address or address ranges.



NOTE: Do not overlap NAT addresses for source NAT, destination NAT, and static NAT within one routing instance.

- Routing instance to which the pool belongs (the default is the main **inet.0** routing instance).
- No port translation (optional)—By default, port address translation is performed with source NAT. If you specify the **port no-translation** option, the number of hosts that the source NAT pool can support is limited to the number of addresses in the pool.
- Overflow pool (optional)—Packets are dropped if there are no addresses available in the designated source NAT pool. To prevent that from happening when the **port no-translation** option is configured, you can specify an overflow pool. Once addresses from the original source NAT pool are exhausted, IP addresses and port numbers are allocated from the overflow pool. A user-defined source NAT pool or an egress interface

can be used as the overflow pool. (When the overflow pool is used, the pool ID is returned with the address.)

- IP address shifting (optional)—A range of original source IP addresses can be mapped to another range of IP addresses by shifting the IP addresses. Specify the **host-address-base** option with the base address of the original source IP address range.

When the **raise-threshold** option is configured for source NAT, an SNMP trap is triggered if the source NAT pool utilization rises above this threshold. If the optional **clear-threshold** option is configured, an SNMP trap is triggered if the source NAT pool utilization drops below this threshold. If **clear-threshold** is not configured it is set by default to 80 percent of the **raise-threshold** value.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Source NAT on page 960
- Source NAT Configuration Overview on page 965
- Understanding Source NAT Pools with PAT on page 962
- Understanding Source NAT Pools Without PAT on page 963
- Understanding Source NAT Pools with Address Shifting on page 963
- Understanding Persistent Addresses on page 964

Understanding Source NAT Pools with PAT

Using the source pool with Port Address Translation (PAT), Junos OS translates both the source IP address and the port number of the packets. When PAT is used, multiple hosts can share the same IP address.

Junos OS maintains a list of assigned port numbers to distinguish what session belongs to which host. When PAT is enabled, up to 64,500 hosts can share a single IP address. Each source pool can contain multiple IP addresses, multiple IP address ranges, or both. For a source pool with PAT, Junos OS may assign different addresses to a single host for different concurrent sessions, unless the source pool or Junos OS has the persistent address feature enabled.

For interface source pool and source pool with PAT, range (1024, 65535) is available for port number mapping per IP address. Within range (1024, 63487) one port is allocated at a time. In range (63488, 65535), two ports are allocated at a time for RTP/RTCP applications such as SIP, H.323, and RTSP.

When a host initiates several sessions that match a policy that requires network address translation and is assigned an address from a source pool that has PAT enabled, the device assigns a different source IP address for each session. Such random address assignment can be problematic for services that create multiple sessions that require the same source IP address for each session. For example, it is important to have the same IP address for multiple sessions when using the AOL Instant Message (AIM) client.

To ensure that the router assigns the same IP address from a source pool to a host for multiple concurrent sessions, you can enable a persistent IP address per router.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Example: Configuring Source NAT Pools (CLI)
 - Example: Configuring a Persistent Address (CLI)
 - Understanding Source NAT on page 960
 - Understanding Source NAT Pools on page 961

Understanding Source NAT Pools Without PAT

When you define a source pool, Junos OS enables PAT by default. To disable PAT, you must specify no port translation when you are defining a source pool.

When using a source pool without PAT, Junos OS performs source Network Address Translation for the IP address without performing PAT for the source port number. For applications that require that a particular source port number remain fixed, you must use source pool without PAT.

The source pool can contain multiple IP addresses, multiple IP address ranges, or both. For source pool without PAT, Junos OS assigns one translated source address to the same host for all its concurrent sessions.

Pool utilization for each source pool without PAT is computed. You can turn on pool utilization alarm by configuring alarm thresholds. An SNMP trap is triggered every time pool utilization rises above a threshold and goes below a threshold.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Example: Configuring Source NAT Pools (CLI)
 - Understanding Source NAT on page 960
 - Understanding Source NAT Pools on page 961

Understanding Source NAT Pools with Address Shifting

The match conditions for a source NAT rule set do not allow you to specify an address range; only address prefixes may be specified in a rule. When configuring a source NAT pool, you can specify the **host-base-address** option; this option specifies the IP address where the original source IP address range begins.

The range of original source IP addresses that are translated is determined by the number of addresses in the source NAT pool. For example, if the source NAT pool contains a range of ten IP addresses, then up to ten original source IP addresses can be translated, starting with a specified base address. This type of translation is one-to-one, static, and without port address translation.

The match condition in a source NAT rule may define a larger address range than that specified in the source NAT pool. For example, a match condition might specify an address prefix that contains 256 addresses, but the source NAT pool contains a range of only ten IP addresses. A packet's source IP address can match a source NAT rule, but if the source IP address is not within the address range specified in the source NAT pool, the source IP address is not translated.

- Related Topics**
- Junos OS Feature Support Reference for SRX Series and J Series Devices
 - Understanding Source NAT
 - Understanding Source NAT Pools
 - Example: Configuring Source NAT with Address Shifting

Understanding Persistent Addresses

By default, port address translation is performed with source NAT. However, an original source address may not be translated to the same IP address for different traffic that originates from the same host. The source NAT **address-persistent** option ensures that the same IP address is assigned from the source NAT pool to a specific host for multiple concurrent sessions.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Source NAT on page 960
 - Source NAT Configuration Overview on page 965
 - Understanding Source NAT Pools with PAT on page 962
 - Example: Configuring Source NAT Pools (CLI)

Understanding Source NAT Rules

Source NAT rules specify two layers of match conditions:

- Traffic direction—Allows you to specify combinations of **from interface**, **from zone**, or **from routing-instance** and **to interface**, **to zone**, or **to routing-instance**. You cannot configure the same **from** and **to** contexts for different rule sets.
- Packet information—Can be source and destination IP addresses or subnets.

If multiple source NAT rules overlap in the match conditions, the most specific rule is chosen. For example, if rules A and B specify the same source and destination IP addresses, but rule A specifies traffic from zone 1 to zone 2 and rule B specifies traffic from zone 1 to interface **ge-0/0/0**, rule B is used to perform source NAT. An interface match is considered to be more specific than a zone match, which is more specific than a routing instance match. For more information about rule set matching, see “Understanding NAT Rule Sets and Rules” on page 928.

The actions you can specify for a source NAT rule are:

- **off**—Do not perform source NAT.
- **pool**—Use the specified user-defined address pool to perform source NAT.
- **interface**—Use the egress interface's IP address to perform source NAT.

Source NAT rules are applied to traffic in the first packet that is processed for the flow or in the fast path for the ALG. Source NAT rules are processed after static NAT rules, destination NAT rules, and reverse mapping of static NAT rules and after route and security policy lookup.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Source NAT on page 960
 - Source NAT Configuration Overview on page 965
 - Understanding NAT Rule Sets and Rules on page 928

Source NAT Configuration Overview

The main configuration tasks for source NAT are as follows:

1. Configure a source NAT address pool that aligns with your network and security requirements (not needed for interface NAT).
2. Configure pool utilization alarms (optional)—Specify thresholds for pool utilization.
3. Configure address persistent (optional)—Ensures that the same IP address is assigned from the source NAT pool to a host for multiple concurrent sessions.
4. Configure source NAT rules that align with your network and security requirements.
5. Configure NAT proxy ARP entries for IP addresses in the same subnet of the ingress interface.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Example: Configuring Source NAT for Egress Interface Translation on page 966
 - Example: Configuring Source NAT for Single Address Translation on page 969
 - Example: Configuring Source NAT for Multiple Addresses with PAT on page 974
 - Example: Configuring Source NAT for Multiple Addresses without PAT on page 979
 - Example: Configuring Source NAT with Address Shifting on page 984
 - Example: Configuring Source NAT with Multiple Rules on page 989
 - Example: Configuring Source and Destination NAT Translations on page 996
 - Verifying NAT Configuration on page 1009

Source NAT Configuration Examples

- Example: Configuring Source NAT for Egress Interface Translation on page 966
- Example: Configuring Source NAT for Single Address Translation on page 969
- Example: Configuring Source NAT for Multiple Addresses with PAT on page 974
- Example: Configuring Source NAT for Multiple Addresses without PAT on page 979
- Example: Configuring Source NAT with Address Shifting on page 984
- Example: Configuring Source NAT with Multiple Rules on page 989
- Example: Configuring Source and Destination NAT Translations on page 996

Example: Configuring Source NAT for Egress Interface Translation

This example describes how to configure a source NAT mapping of private addresses to the public address of an egress interface.

- Requirements on page 966
- Overview on page 966
- Configuration on page 967
- Verification on page 969

Requirements

Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 87.

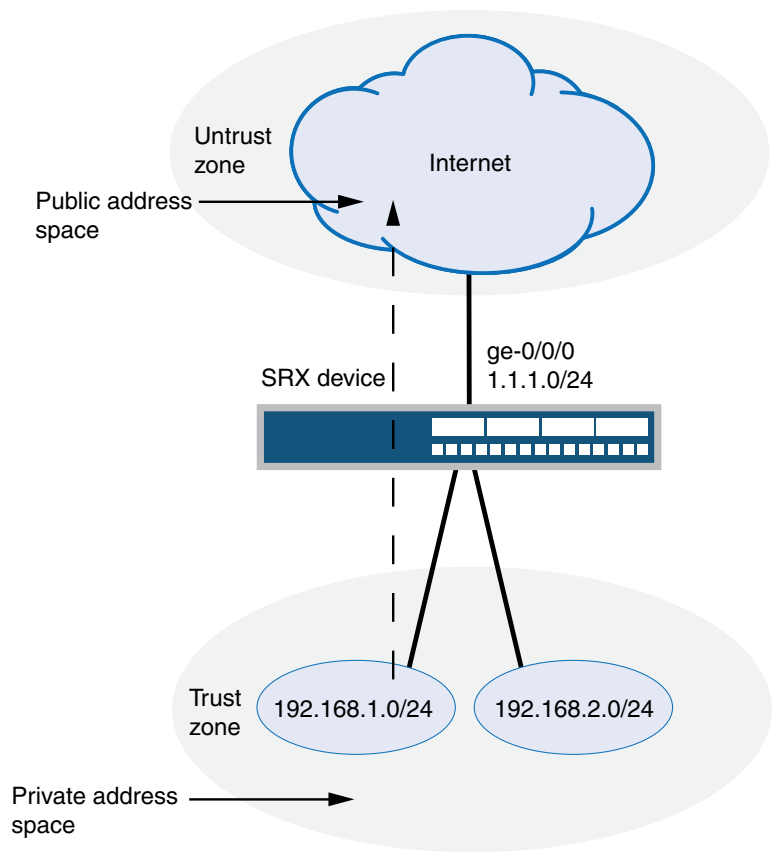
Overview

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 114 on page 967, devices with private addresses in the trust zone access a public network through the egress interface ge-0/0/0. For packets that enter the Juniper Networks security device from the trust zone with a destination address in the untrust zone, the source IP address is translated to the IP address of the egress interface.



NOTE: No source NAT pool is required for source NAT using an egress interface. Proxy ARP does not need to be configured for the egress interface.

Figure 114: Source NAT Egress Interface Translation



Original Source IP	Translated Source IP
0.0.0.0/0	1.1.1.63 (Interface IP)

g030668

This example describes the following configurations:

- Source NAT rule set **rs1** with a rule **r1** to match any packet from the trust zone to the untrust zone. For matching packets, the source address is translated to the IP address of the egress interface.
- Security policies to permit traffic from the trust zone to the untrust zone.

Configuration

CLI Quick Configuration To quickly configure a source NAT mapping to an egress interface, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
```

```

set security nat source rule-set rs1 rule r1 match source-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 then source-nat interface
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit

```

Step-by-Step Procedure

The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see [Using the CLI Editor in Configuration Mode](#).

To configure a source NAT translation to an egress interface:

1. Create a source NAT rule set.

```

[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust

```
2. Configure a rule that matches packets and translates the source address to the address of the egress interface.

```

[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 then source-nat interface

```
3. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```

[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
  destination-address any application any
user@host# set policy internet-access then permit

```

Results From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security nat
source {
  rule-set rs1 {
    from zone trust;
    to zone untrust;
    rule r1 {
      match {
        source-address 0.0.0.0/0;
        destination-address 0.0.0.0/0;
      }
      then {
        source-nat {

```



```

        interface;
    }
}
}
}
user@host# show security policies
from-zone trust to-zone untrust {
  policy internet-access {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Rule Usage on page 969
- Verifying NAT Application to Traffic on page 969

Verifying Source NAT Rule Usage

Purpose Verify that there is traffic matching the source NAT rule.

Action From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

Verifying NAT Application to Traffic

Purpose Verify that NAT is being applied to the specified traffic.

Action From operational mode, enter the **show security flow session** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Source NAT on page 960
 - Source NAT Configuration Overview on page 965

Example: Configuring Source NAT for Single Address Translation

This example describes how to configure a source NAT mapping of a single private address to a public address.

- Requirements on page 970
- Overview on page 970

- Configuration on page 972
- Verification on page 974

Requirements

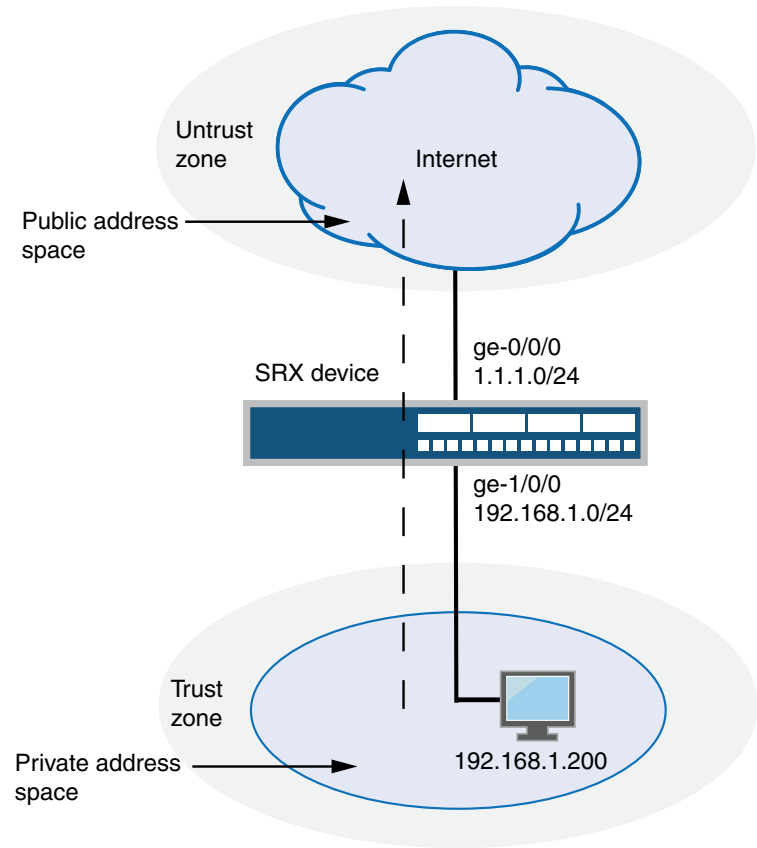
Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 87.

Overview

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 115 on page 971, a device with the private address 192.168.1.200 in the trust zone accesses a public network. For packets sent by the device to a destination address in the untrust zone, the Juniper Networks security device translates the source IP address to the public IP address 1.1.1.200/32.

Figure 115: Source NAT Single Address Translation



Original Source IP	Translated Source IP
192.168.1.200/32	1.1.1.200/32

g030669

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address 1.1.1.200/32.
- Source NAT rule set **rs1** with rule **r1** to match packets from the trust zone to the untrust zone with the source IP address 192.168.1.200/32. For matching packets, the source address is translated to the IP address in **src-nat-pool-1** pool.
- Proxy ARP for the address 1.1.1.200 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for that address.
- Security policies to permit traffic from the trust zone to the untrust zone.

Configuration

CLI Quick Configuration

To quickly configure a source NAT mapping for a single IP address, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 1.1.1.200/32
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 192.168.1.200/32
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.200/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
```

Step-by-Step Procedure

The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a source NAT translation for a single IP address:

1. Create a source NAT pool.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 address 1.1.1.200/32
```
2. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust
```
3. Configure a rule that matches packets and translates the source address to the address in the pool.

```
[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address 192.168.1.200/32
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
```
4. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.200
```
5. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
  destination-address any application any
user@host# set policy internet-access then permit
```

[edit]

```
source {
    pool src-nat-pool-1 {
        address {
            1.1.1.200/32;
        }
    }
}
rule-set rs1 {
    from zone trust;
    to zone untrust;
    rule r1 {
        match {
            source-address 192.168.1.200/32;
        }
        then {
            source-nat {
                pool {
                    src-nat-pool-1;
                }
            }
        }
    }
}
}
}
proxy-arp {
    interface ge-0/0/0.0 {
        address {
            1.1.1.200/32;
        }
    }
}
user@host# show security policies
from-zone trust to-zone untrust {
    policy internet-access {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
```

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 974
- Verifying Source NAT Rule Usage on page 974
- Verifying NAT Application to Traffic on page 974

Verifying Source NAT Pool Usage

Purpose Verify that there is traffic using IP addresses from the source NAT pool.

Action From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

Verifying Source NAT Rule Usage

Purpose Verify that there is traffic matching the source NAT rule.

Action From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

Verifying NAT Application to Traffic

Purpose Verify that NAT is being applied to the specified traffic.

Action From operational mode, enter the **show security flow session** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Source NAT on page 960
- Source NAT Configuration Overview on page 965

Example: Configuring Source NAT for Multiple Addresses with PAT

This example describes how to configure a source NAT mapping of a private address block to a smaller public address block using port address translation.

- Requirements on page 974
- Overview on page 975
- Configuration on page 977
- Verification on page 979

Requirements

Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 87.

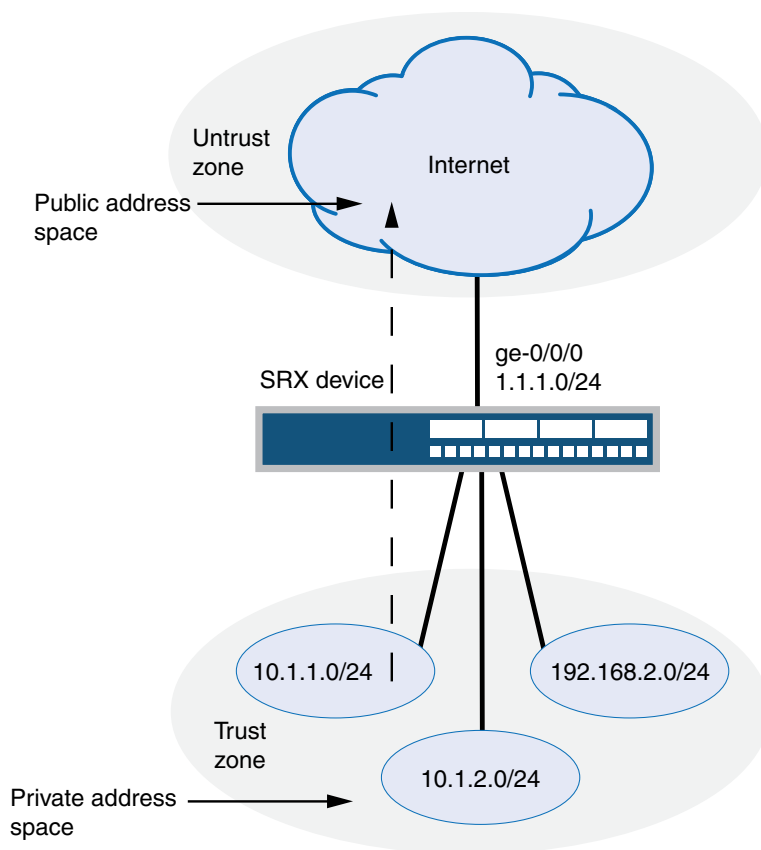
Overview

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 116 on page 976, the source IP address in packets sent from the trust zone to the untrust zone is mapped to a smaller block of public addresses in the range from 1.1.1.1/32 through 1.1.1.24/32. Because the size of the source NAT address pool is smaller than the number of potential addresses that might need to be translated, port address translation is used.



NOTE: Port address translation includes a source port number with the source IP address mapping. This allows multiple addresses on a private network to map to a smaller number of public IP addresses. Port address translation is enabled by default for source NAT pools.

Figure 116: Source NAT Multiple Addresses with PAT



Original Source IP	Translated Source IP
10.1.1.0/24	1.1.1.1 (with port address translation)
10.1.2.0/24	
192.168.1.0/24	

g030670

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address range 1.1.1.1/32 through 1.1.1.24/32.
- Source NAT rule set **rs1** to match all packets from the trust zone to the untrust zone. For matching packets, the source IP address is translated to an IP address in the **src-nat-pool-1** pool.

- Proxy ARP for the addresses 1.1.1.1/32 through 1.1.1.24/32 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses.
- Security policies to permit traffic from the trust zone to the untrust zone.

Configuration

CLI Quick Configuration To quickly configure a source NAT mapping from a private address block to a smaller public address block using PAT, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 1.1.1.1/32 to 1.1.1.24/32
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 10.1.1.0/24
set security nat source rule-set rs1 rule r1 match source-address 10.1.2.0/24
set security nat source rule-set rs1 rule r1 match source-address 192.168.1.0/24
set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.1/32 to 1.1.1.24/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
```

Step-by-Step Procedure The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode.

To configure a source NAT mapping from a private address block to a smaller public address block using PAT:

1. Create a source NAT pool.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 address 1.1.1.1 to 1.1.1.24
```
2. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust
```
3. Configure a rule that matches packets and translates the source address to an address in the pool.

```
[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address [10.1.1.0/24 10.1.2.0/24
  192.168.1.0/24]
user@host# set rule-set rs1 rule r1 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
```

4. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.1 to 1.1.1.24
```

5. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
destination-address any application any
user@host# set policy internet-access then permit
```

Results From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool src-nat-pool-1 {
    address {
      1.1.1.1/32 to 1.1.1.24/32;
    }
  }
}
rule-set rs1 {
  from zone trust;
  to zone untrust;
  rule r1 {
    match {
      source-address [10.1.1.0/24 10.1.2.0/24 192.168.1.0/24];
      destination-address 0.0.0.0/0;
    }
    then {
      source-nat {
        pool {
          src-nat-pool-1;
        }
      }
    }
  }
}
}
}
proxy-arp {
  interface ge-0/0/0.0 {
    address {
      1.1.1.1/32 to 1.1.1.24/32;
    }
  }
}
}
user@host# show security policies
from-zone trust to-zone untrust {
  policy internet-access {
    match {
      source-address any;
      destination-address any;
    }
  }
}
```

```

        application any;
    }
    then {
        permit;
    }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 979
- Verifying Source NAT Rule Usage on page 979
- Verifying NAT Application to Traffic on page 979

Verifying Source NAT Pool Usage

Purpose Verify that there is traffic using IP addresses from the source NAT pool.

Action From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

Verifying Source NAT Rule Usage

Purpose Verify that there is traffic matching the source NAT rule.

Action From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

Verifying NAT Application to Traffic

Purpose Verify that NAT is being applied to the specified traffic.

Action From operational mode, enter the **show security flow session** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Source NAT on page 960
 - Source NAT Configuration Overview on page 965
 - Understanding Source NAT Pools with PAT on page 962

Example: Configuring Source NAT for Multiple Addresses without PAT

This example describes how to configure a source NAT mapping of a private address block to a smaller public address block without port address translation.



NOTE: Port address translation is enabled by default for source NAT pools. When port address translation is disabled, the number of translations that the source NAT pool can concurrently support is limited to the number of addresses in the pool. Packets are dropped if there are no addresses available in the source NAT pool. You can optionally specify an overflow pool from which IP addresses and port numbers are allocated when there are no addresses available in the original source NAT pool.

- Requirements on page 980
- Overview on page 980
- Configuration on page 982
- Verification on page 984

Requirements

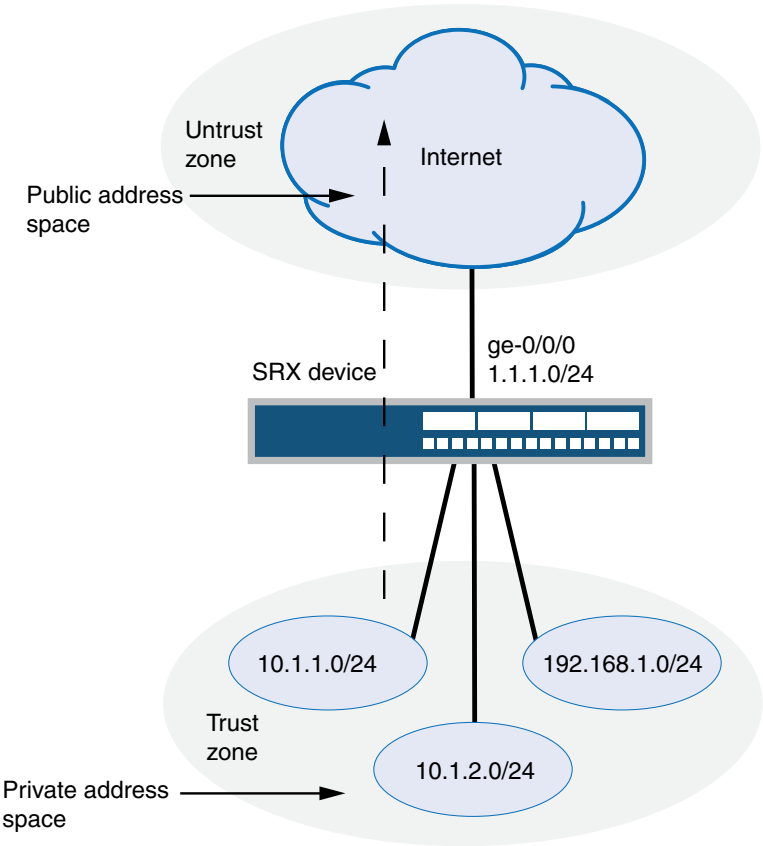
Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 87.

Overview

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 117 on page 981, the source IP address in packets sent from the trust zone to the untrust zone is mapped to a smaller block of public addresses in the range from 1.1.1.1/32 through 1.1.1.24/32.

Figure 117: Source NAT Multiple Addresses without PAT



Original Source IP	Translated Source IP
10.1.1.0/24 10.1.2.0/24 192.168.1.0/24	1.1.1.1 (no port address translation)

g030671

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address range 1.1.1.1/32 through 1.1.1.24/32. The **port no-translation** option is specified for the pool.
- Source NAT rule set **rs1** to match all packets from the trust zone to the untrust zone. For matching packets, the source IP address is translated to an IP address in the **src-nat-pool-1** pool.

- Proxy ARP for the addresses 1.1.1.1/32 through 1.1.1.24/32 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses.
- Security policies to permit traffic from the trust zone to the untrust zone.

Configuration

CLI Quick Configuration To quickly configure a source NAT mapping from a private address block to a smaller public address block without PAT, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 1.1.1.1/32 to 1.1.1.24/32
set security nat source pool src-nat-pool-1 port no-translation
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.1/32 to 1.1.1.24/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
```

Step-by-Step Procedure The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure a source NAT mapping from a private address block to a smaller public address block without PAT:

1. Create a source NAT pool.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 address 1.1.1.1 to 1.1.1.24
```
2. Specify the **port no-translation** option.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 port no-translation
```
3. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust
```
4. Configure a rule that matches packets and translates the source address to an address in the pool.

```
[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address 0.0.0.0/0
```

```

user@host# set rule-set rs1 rule r1 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1

```

5. Configure proxy ARP.

```

[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.1 to 1.1.1.24

```

6. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```

[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
destination-address any application any
user@host# set policy internet-access then permit

```

Results From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show security nat
source {
  pool src-nat-pool-1 {
    address {
      1.1.1.1/32 to 1.1.1.24/32;
    }
    port no-translation;
  }
}
rule-set rs1 {
  from zone trust;
  to zone untrust;
  rule r1 {
    match {
      source-address 0.0.0.0/0;
      destination-address 0.0.0.0/0;
    }
    then {
      source-nat {
        pool {
          src-nat-pool-1;
        }
      }
    }
  }
}
}
}
proxy-arp {
  interface ge-0/0/0.0 {
    address {
      1.1.1.1/32 to 1.1.1.24/32;
    }
  }
}
}
user@host# show security policies
from-zone trust to-zone untrust {

```

```
policy internet-access {  
  match {  
    source-address any;  
    destination-address any;  
    application any;  
  }  
  then {  
    permit;  
  }  
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 984
- Verifying Source NAT Rule Usage on page 984
- Verifying NAT Application to Traffic on page 984

Verifying Source NAT Pool Usage

Purpose Verify that there is traffic using IP addresses from the source NAT pool.

Action From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

Verifying Source NAT Rule Usage

Purpose Verify that there is traffic matching the source NAT rule.

Action From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

Verifying NAT Application to Traffic

Purpose Verify that NAT is being applied to the specified traffic.

Action From operational mode, enter the **show security flow session** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Source NAT on page 960
 - Source NAT Configuration Overview on page 965
 - Understanding Source NAT Pools Without PAT on page 963

Example: Configuring Source NAT with Address Shifting

This example describes how to configure a source NAT mapping of a private address range to public addresses, with optional address shifting. This mapping is one-to-one

between the original source IP addresses and translated IP addresses and no port translation is performed.



NOTE: The match conditions for a source NAT rule set do not allow you to specify an address range; only address prefixes may be specified in a rule. When configuring a source NAT pool, you can specify the `host-base-address` option; this option specifies the IP address where the original source IP address range begins.

The range of original source IP addresses that are translated is determined by the number of addresses in the source NAT pool. For example, if the source NAT pool contains a range of ten IP addresses, then up to ten original source IP addresses can be translated, starting with a specified base address.

The match condition in a source NAT rule may define a larger address range than that specified in the source NAT pool. For example, a match condition might specify an address prefix that contains 256 addresses, but the source NAT pool contains a range of only ten IP addresses. A packet's source IP address can match a source NAT rule, but if the source IP address is not within the address range specified in the source NAT pool, the source IP address is not translated.

- Requirements on page 985
- Overview on page 985
- Configuration on page 987
- Verification on page 989

Requirements

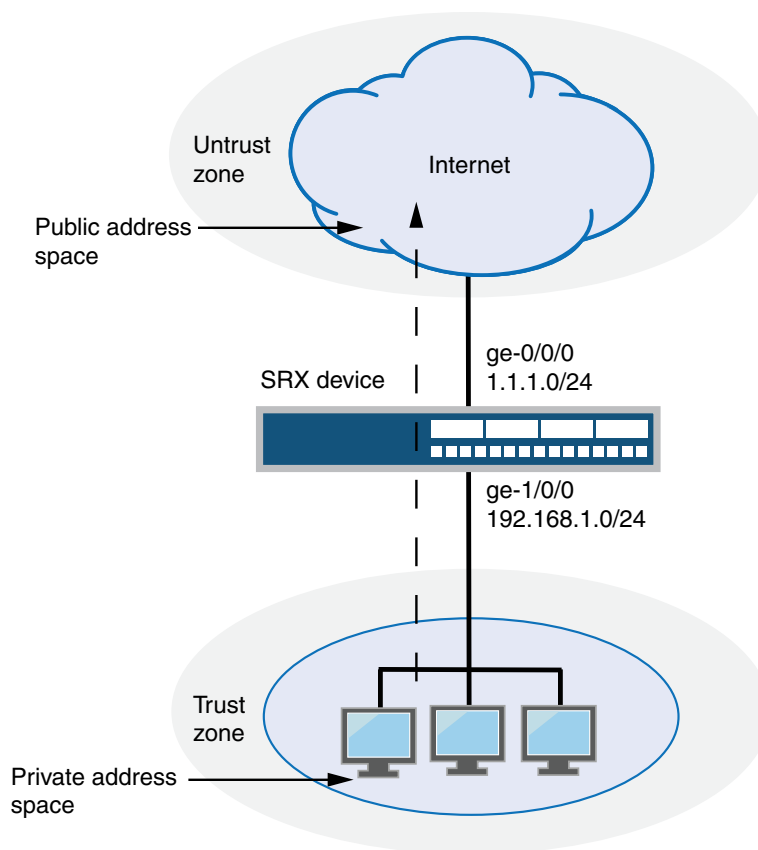
Before you begin:

1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See "Understanding Security Zones" on page 87.

Overview

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 118 on page 986, a range of private addresses in the trust zone is mapped to a range of public addresses in the untrust zone. For packets sent from the trust zone to the untrust zone, a source IP address in the range of 192.168.1.10/32 through 192.168.1.20/32 is translated to a public address in the range of 1.1.1.30/32 through 1.1.1.40/32.

Figure 118: Source NAT with Address Shifting



Original Source IP	Translated Source IP
192.168.1.10/32 - 192.168.1.20/32	1.1.1.30/32 - 1.1.1.40/32

g030672

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address range 1.1.1.30/32 through 1.1.1.40/32. For this pool, the beginning of the original source IP address range is 192.168.1.10/32 and is specified with the **host-address-base** option.
- Source NAT rule set **rs1** with rule **r1** to match packets from the trust zone to the untrust zone with a source IP address in the 192.168.1.0/24 subnet. For matching packets that fall within the source IP address range specified by the **src-nat-pool-1** configuration, the source address is translated to the IP address in **src-nat-pool-1** pool.

- Proxy ARP for the addresses 1.1.1.30/32 through 1.1.1.40/32 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for that address.
- Security policies to permit traffic from the trust zone to the untrust zone.

Configuration

CLI Quick Configuration To quickly configure a source NAT mapping with address shifting, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 1.1.1.30/32 to 1.1.1.40/32
set security nat source pool src-nat-pool-1 host-address-base 192.168.1.10/32
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 192.168.1.0/24
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.30/32 to 1.1.1.40/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
```

Step-by-Step Procedure The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode.

To configure a source NAT mapping with address shifting:

1. Create a source NAT pool.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 address 1.1.1.30/32 to 1.1.1.40/32
```
2. Specify the beginning of the original source IP address range.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 host-address-base 192.168.1.10/32
```
3. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust
```
4. Configure a rule that matches packets and translates the source address to an address in the pool.

```
[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address 192.168.1.0/24
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
```
5. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.30/32 to 1.1.1.40/32
```

6. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
destination-address any application any
user@host# set policy internet-access then permit
```

Results From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
    pool src-nat-pool-1 {
        address {
            1.1.1.30/32 to 1.1.1.40/32;
        }
        host-address-base 192.168.1.10/32;
    }
}
rule-set rs1 {
    from zone trust;
    to zone untrust;
    rule r1 {
        match {
            source-address 192.168.1.0/24;
        }
        then {
            source-nat {
                pool {
                    src-nat-pool-1;
                }
            }
        }
    }
}
}
}
proxy-arp {
    interface ge-0/0/0.0 {
        address {
            1.1.1.30/32 to 1.1.1.40/32;
        }
    }
}
}
user@host# show security policies
from-zone trust to-zone untrust {
    policy internet-access {
        match {
            source-address any;
            destination-address any;
            application any;
```

```

    }
    then {
        permit;
    }
}
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 989
- Verifying Source NAT Rule Usage on page 989
- Verifying NAT Application to Traffic on page 989

Verifying Source NAT Pool Usage

Purpose Verify that there is traffic using IP addresses from the source NAT pool.

Action From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

Verifying Source NAT Rule Usage

Purpose Verify that there is traffic matching the source NAT rule.

Action From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

Verifying NAT Application to Traffic

Purpose Verify that NAT is being applied to the specified traffic.

Action From operational mode, enter the **show security flow session** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Source NAT on page 960
 - Source NAT Configuration Overview on page 965
 - Understanding Source NAT Pools with Address Shifting on page 963

Example: Configuring Source NAT with Multiple Rules

This example describes how to configure source NAT mappings with multiple rules.

- Requirements on page 990
- Overview on page 990
- Configuration on page 992
- Verification on page 996

Requirements

Before you begin:

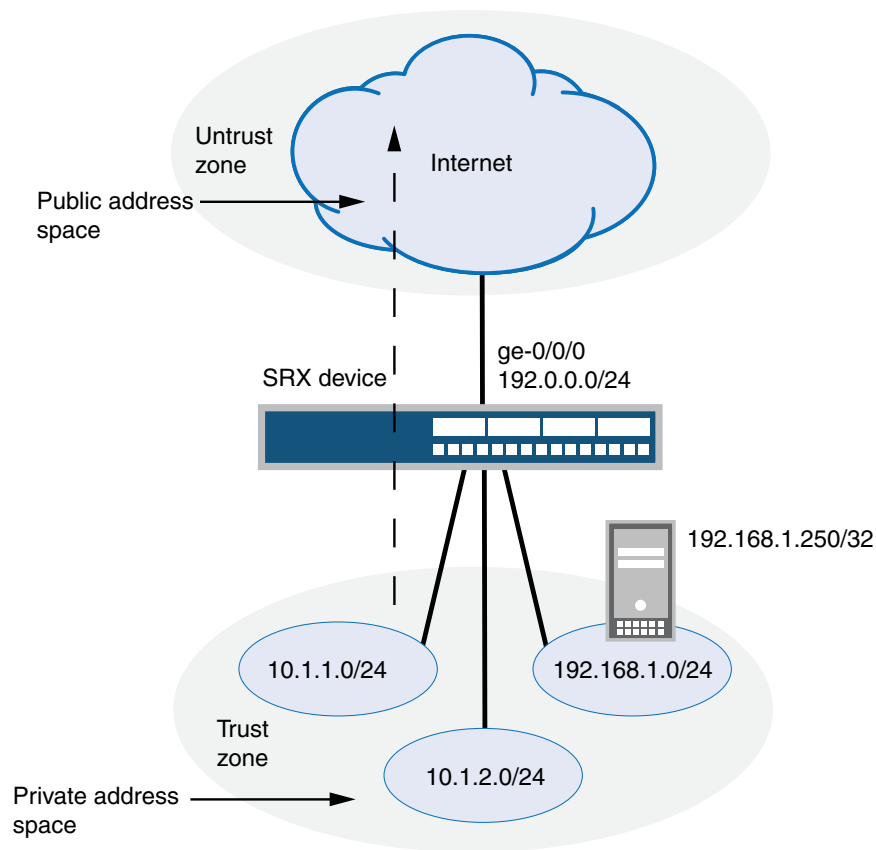
1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See “Understanding Security Zones” on page 87.

Overview

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 119 on page 991, the following translations are performed on the Juniper Networks security device for the source NAT mapping for traffic from the trust zone to the untrust zones:

- The source IP address in packets sent by the 10.1.1.0/24 and 10.1.2.0/24 subnets to any address in the untrust zone is translated to a public address in the range from 192.0.0.1 to 192.0.0.24 with port translation.
- The source IP address in packets sent by the 192.168.1.0/24 subnet to any address in the untrust zone is translated to a public address in the range from 192.0.0.100 to 192.0.0.249 with no port translation.
- The source IP address in packets sent by the 192.168.1.250/32 host device is not translated.

Figure 119: Source NAT with Multiple Translation Rules



Original Source IP	Translated Source IP
10.1.1.0/24, 10.1.2.0/24	192.0.0.1 - 192.0.0.24 (w/port translation)
192.168.1.0/24	192.0.0.100 - 192.0.0.249 (no port translation)
192.168.1.250/32	(no source NAT translation)

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address range 192.0.0.1 through 192.0.0.24.
- Source NAT pool **src-nat-pool-2** that contains the IP address range 192.0.0.100 through 192.0.0.249, with port address translation disabled.



NOTE: When port address translation is disabled, the number of translations that the source NAT pool can support concurrently is limited to the number of addresses in the pool. Packets are dropped if there are no addresses available in the source NAT pool. You can optional specify an overflow pool from which IP addresses and port numbers are allocated when there are no addresses available in the original source NAT pool.

- Source NAT rule set **rs1** to match packets from the trust zone to the untrust zone. Rule set **rs1** contains multiple rules:
 - Rule **r1** to match packets with a source IP address in either the 10.1.1.0/24 or 10.1.2.0/24 subnets. For matching packets, the source address is translated to an IP address in the **src-nat-pool-1** pool.
 - Rule **r2** to match packets with a source IP address of 192.168.1.250/32. For matching packets, there is no NAT translation performed.
 - Rule **r3** to match packets with a source IP address in the 192.168.1.0/24 subnet. For matching packets, the source address is translated to an IP address in the **src-nat-pool-2** pool.



NOTE: The order of rules in a rule set is important, as the first rule in the rule set that matches the traffic is used. Therefore, rule **r2** to match a specific IP address must be placed before rule **r3** that matches the subnet on which the device is located.

- Proxy ARP for the addresses 192.0.0.1 through 192.0.0.24 and 192.0.0.100 through 192.0.0.249 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses.
- Security policies to permit traffic from the trust zone to the untrust zone.

Configuration

CLI Quick Configuration

To quickly configure a source NAT mapping with multiple rules, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 192.0.0.1/32 to 192.0.0.24/32
set security nat source pool src-nat-pool-2 address 192.0.0.100/32 to 192.0.0.249/32
set security nat source pool src-nat-pool-2 port no-translation
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 10.1.1.0/24
set security nat source rule-set rs1 rule r1 match source-address 10.1.2.0/24
set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat source rule-set rs1 rule r2 match source-address 192.168.1.250/32
set security nat source rule-set rs1 rule r2 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r2 then source-nat off
set security nat source rule-set rs1 rule r3 match source-address 192.168.1.0/24
set security nat source rule-set rs1 rule r3 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r3 then source-nat pool src-nat-pool-2
```



```

set security nat proxy-arp interface ge-0/0/0.0 address 192.0.0.1/32 to 192.0.0.24/32
set security nat proxy-arp interface ge-0/0/0.0 address 192.0.0.100/32 to 192.0.0.249/32
set security policies from-zone trust to-zone untrust policy internet-access match
  source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
  application any
set security policies from-zone trust to-zone untrust policy internet-access then permit

```

Step-by-Step Procedure

The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure multiple source NAT rules in a rule set:

1. Create a source NAT pool.

```

[edit security nat source]
user@host# set pool src-nat-pool-1 address 192.0.0.1 to 192.0.0.24

```
2. Create a source NAT pool with no port translation.

```

[edit security nat source]
user@host# set pool src-nat-pool-2 address 192.0.0.100 to 192.0.0.249
user@host# set pool src-nat-pool-2 port no-translation

```



NOTE: To configure an overflow pool for src-nat-pool-2 using the egress interface:

```

[edit security nat source]
user@host# set pool src-nat-pool-2 overflow-pool interface

```

3. Create a source NAT rule set.

```

[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust

```
4. Configure a rule that matches packets and translates the source address to an address in the pool.

```

[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address [10.1.1.0/24 10.1.2.0/24]
user@host# set rule-set rs1 rule r1 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1

```
5. Configure a rule to match packets for which the source address is not translated.

```

[edit security nat source]
user@host# set rule-set rs1 rule r2 match source-address 192.168.1.250/32
user@host# set rule-set rs1 rule r2 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r2 then source-nat off

```

6. Configure a rule to match packets and translate the source address to an address in the pool with no port translation.

```
[edit security nat source]
user@host# set rule-set rs1 rule r3 match source-address 192.168.1.0/24
user@host# set rule-set rs1 rule r3 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r3 then source-nat pool src-nat-pool-2
```

7. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 192.0.0.1 to 192.0.0.24
user@host# set proxy-arp interface ge-0/0/0.0 address 192.0.0.100 to 192.0.0.249
```

8. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
destination-address any application any
user@host# set policy internet-access then permit
```

Results From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool src-nat-pool-1 {
    address {
      192.0.0.1/32 to 192.0.0.24/32;
    }
  }
  pool src-nat-pool-2 {
    address {
      192.0.0.100/32 to 192.0.0.249/32;
    }
    port no-translation;
  }
  rule-set rs1 {
    from zone trust;
    to zone untrust;
    rule r1 {
      match {
        source-address [ 10.1.1.0/24 10.1.2.0/24 ];
        destination-address 0.0.0.0/0;
      }
      then {
        source-nat {
          pool {
            src-nat-pool-1;
          }
        }
      }
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 996
- Verifying Source NAT Rule Usage on page 996
- Verifying NAT Application to Traffic on page 996

Verifying Source NAT Pool Usage

Purpose Verify that there is traffic using IP addresses from the source NAT pool.

Action From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

Verifying Source NAT Rule Usage

Purpose Verify that there is traffic matching the source NAT rule.

Action From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

Verifying NAT Application to Traffic

Purpose Verify that NAT is being applied to the specified traffic.

Action From operational mode, enter the **show security flow session** command.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Understanding Source NAT on page 960
- Source NAT Configuration Overview on page 965
- Understanding Source NAT Rules on page 964

Example: Configuring Source and Destination NAT Translations

This example describes how to configure both source and destination NAT mappings.

- Requirements on page 996
- Overview on page 997
- Configuration on page 998
- Verification on page 1001

Requirements

Before you begin:

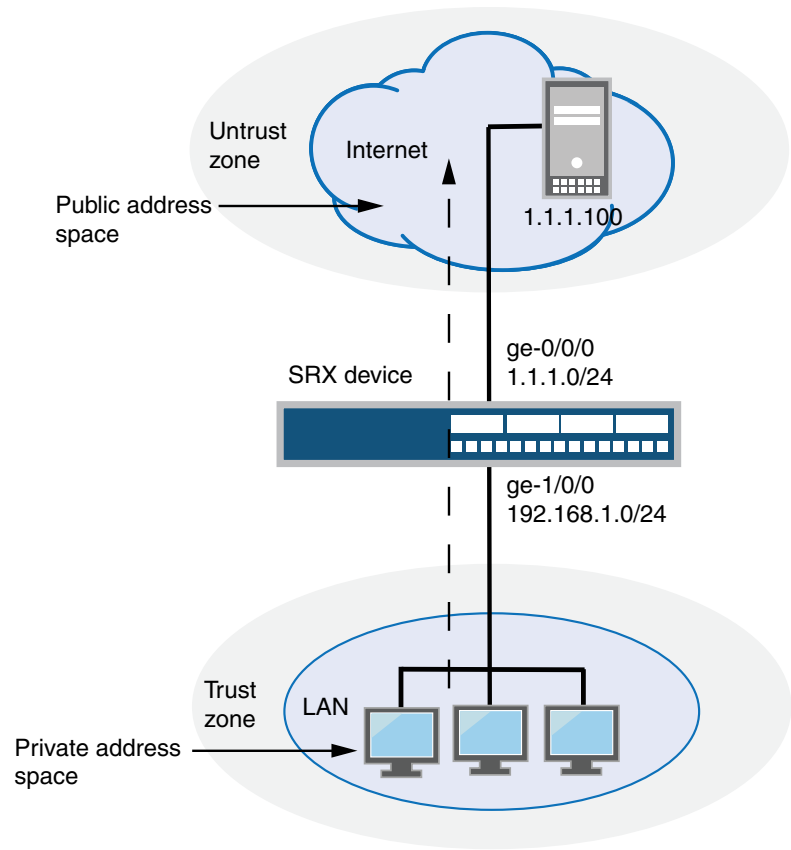
1. Configure network interfaces on the device. See the *Junos OS Interfaces Configuration Guide for Security Devices*.
2. Create security zones and assign interfaces to them. See "Understanding Security Zones" on page 87.

Overview

This example uses the trust security zone for the private address space and the untrust security zone for the public address space. In Figure 120 on page 997, the following translations are performed on the Juniper Networks security device:

- The source IP address in packets sent by the device with the private address 192.168.1.200 in the trust zone to any address in the untrust zone is translated to a public address in the range from 1.1.1.10 through 1.1.1.14.
- The destination IP address 1.1.1.100/32 in packets sent from the trust zone to the untrust zone is translated to the address 10.1.1.200/32.

Figure 120: Source and Destination NAT Translations



Original Source IP 192.168.1..0/24	Translated Source IP 1.1.1.10 - 1.1.1.14
Original Destination IP 1.1.1.100/32	Translated Destination IP 10.1.1.200/32

This example describes the following configurations:

- Source NAT pool **src-nat-pool-1** that contains the IP address range 1.1.1.10 through 1.1.1.14.
- Source NAT rule set **rs1** with rule **r1** to match any packets from the trust zone to the untrust zone. For matching packets, the source address is translated to an IP address in the **src-nat-pool-1** pool.
- Destination NAT pool **dst-nat-pool-1** that contains the IP address 10.1.1.200/32.
- Destination NAT rule set **rs1** with rule **r1** to match packets from the trust zone with the destination IP address 1.1.1.100. For matching packets, the destination address is translated to the IP address in the **dst-nat-pool-1** pool.
- Proxy ARP for the addresses 1.1.1.10 through 1.1.1.14 and 1.1.1.100/32 on interface ge-0/0/0.0. This allows the Juniper Networks security device to respond to ARP requests received on the interface for those addresses.
- Security policy to permit traffic from the trust zone to the untrust zone.
- Security policy to permit traffic from the untrust zone to the translated destination IP addresses in the trust zone.

Configuration

CLI Quick Configuration

To quickly configure source and destination NAT mappings, copy the following commands and paste them into the CLI.

```
[edit]
set security nat source pool src-nat-pool-1 address 1.1.1.10/32 to 1.1.1.14/32
set security nat source rule-set rs1 from zone trust
set security nat source rule-set rs1 to zone untrust
set security nat source rule-set rs1 rule r1 match source-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 match destination-address 0.0.0.0/0
set security nat source rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
set security nat destination pool dst-nat-pool-1 address 10.1.1.200/32
set security nat destination rule-set rs1 from zone untrust
set security nat destination rule-set rs1 rule r1 match destination-address 1.1.1.100/32
set security nat destination rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.10/32 to 1.1.1.24/32
set security nat proxy-arp interface ge-0/0/0.0 address 1.1.1.100/32
set security policies from-zone trust to-zone untrust policy internet-access match
    source-address any
set security policies from-zone trust to-zone untrust policy internet-access match
    destination-address any
set security policies from-zone trust to-zone untrust policy internet-access match
    application any
set security policies from-zone trust to-zone untrust policy internet-access then permit
set security zones security-zone trust address-book address dst-nat-pool-1 10.1.1.200/32
set security policies from-zone untrust to-zone trust policy dst-nat-pool-1-access match
    source-address any
set security policies from-zone untrust to-zone trust policy dst-nat-pool-1-access match
    destination-address dst-nat-pool-1
set security policies from-zone untrust to-zone trust policy dst-nat-pool-1-access match
    application any
```

```
set security policies from-zone untrust to-zone trust policy dst-nat-pool-1-access then
permit
```

**Step-by-Step
Procedure**

The following example requires you to navigate throughout various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode*.

To configure the source and destination NAT translations:

1. Create a source NAT pool.

```
[edit security nat source]
user@host# set pool src-nat-pool-1 address 1.1.1.10 to 1.1.1.14
```
2. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set rs1 from zone trust
user@host# set rule-set rs1 to zone untrust
```
3. Configure a rule that matches packets and translates the source address to an address in the source NAT pool.

```
[edit security nat source]
user@host# set rule-set rs1 rule r1 match source-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 match destination-address 0.0.0.0/0
user@host# set rule-set rs1 rule r1 then source-nat pool src-nat-pool-1
```
4. Create a destination NAT pool.

```
[edit security nat destination]
user@host# set pool dst-nat-pool-1 address 10.1.1.200/32
```
5. Create a destination NAT rule set.

```
[edit security nat destination]
user@host# set rule-set rs1 from zone untrust
```
6. Configure a rule that matches packets and translates the destination address to the address in the destination NAT pool.

```
[edit security nat destination]
user@host# set rule-set rs1 rule r1 match destination-address 1.1.1.100/32
user@host# set rule-set rs1 rule r1 then destination-nat pool dst-nat-pool-1
```
7. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.10 to 1.1.1.14
user@host# set proxy-arp interface ge-0/0/0.0 address 1.1.1.100
```
8. Configure a security policy that allows traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy internet-access match source-address any
destination-address any application any
user@host# set policy internet-access then permit
```
9. Configure an address book entry in the trust zone for the translated destination IP address.

```
[edit security]
user@host# set zones security-zone trust address-book address dst-nat-pool-1
10.1.1.200/32
```

10. Configure a security policy that allows traffic from the untrust zone to the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy dst-nat-pool-1-access match source-address any
destination-address dst-nat-pool-1 application any
user@host# set policy dst-nat-pool-1-access then permit
```

Results From configuration mode, confirm your configuration by entering the **show security nat** and **show security policies** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
  pool src-nat-pool-1 {
    address {
      1.1.1.10/32 to 1.1.1.14/32;
    }
  }
  rule-set rs1 {
    to zone untrust;
    rule r1 {
      match {
        source-address 0.0.0.0/0;
        destination-address 0.0.0.0/0;
      }
      then {
        source-nat {
          pool {
            src-nat-pool-1;
          }
        }
      }
    }
  }
}
destination {
  pool dst-nat-pool-1 {
    address 10.1.1.200/32;
  }
  rule-set rs1 {
    from zone untrust;
    rule r1 {
      match {
        destination-address 1.1.1.100/32;
      }
      then {
        destination-nat pool dst-nat-pool-1;
      }
    }
  }
}
```



```

    }
  }
  proxy-arp {
    interface ge-0/0/0.0 {
      address {
        1.1.1.10/32 to 1.1.1.24/32;
        1.1.1.100/32;
      }
    }
  }
}
user@host# show security policies
from-zone trust to-zone untrust {
  policy internet-access {
    match {
      source-address any;
      destination-address any;
      application any;
    }
  }
  policy internet-access {
    then {
      permit;
    }
  }
}
from-zone untrust to-zone trust {
  policy dst-nat-pool-1-access {
    match {
      source-address any;
      destination-address dst-nat-pool-1;
      application any;
    }
    then {
      permit;
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying Source NAT Pool Usage on page 1001
- Verifying Source NAT Rule Usage on page 1002
- Verifying Destination NAT Pool Usage on page 1002
- Verifying Destination NAT Rule Usage on page 1002
- Verifying NAT Application to Traffic on page 1002

Verifying Source NAT Pool Usage

Purpose Verify that there is traffic using IP addresses from the source NAT pool.

Action From operational mode, enter the **show security nat source pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

Verifying Source NAT Rule Usage

Purpose Verify that there is traffic matching the source NAT rule.

Action From operational mode, enter the **show security nat source rule all** command. View the Translation hits field to check for traffic that matches the rule.

Verifying Destination NAT Pool Usage

Purpose Verify that there is traffic using IP addresses from the destination NAT pool.

Action From operational mode, enter the **show security nat destination pool all** command. View the Translation hits field to check for traffic using IP addresses from the pool.

Verifying Destination NAT Rule Usage

Purpose Verify that there is traffic matching the destination NAT rule.

Action From operational mode, enter the **show security nat destination rule all** command. View the Translation hits field to check for traffic that matches the rule.

Verifying NAT Application to Traffic

Purpose Verify that NAT is being applied to the specified traffic.

Action From operational mode, enter the **show security flow session** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Source NAT on page 960
 - Source NAT Configuration Overview on page 965
 - Understanding Destination NAT on page 942
 - Destination NAT Configuration Overview on page 944

Disabling Port Randomization for Source NAT (CLI Procedure)

For pool-based source NAT and interface NAT, port numbers are allocated randomly by default. Although randomized port number allocation can provide protection from security threats such as DNS poison attacks, it can also affect performance and memory usage for pool-based source NAT.

You can disable port randomization by using the **port-randomization disable** statement at the **[edit security nat source]** hierarchy level. To re-enable port randomization, use the **port-randomization** statement at the **[edit security nat source]** hierarchy level.

```
user@host# set security nat source port-randomization disable
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Source NAT on page 960
 - Source NAT Configuration Overview on page 965

Persistent NAT

- Understanding Persistent NAT on page 1003
- Understanding Session Traversal Utilities for NAT (STUN) Protocol on page 1004
- Persistent NAT Configuration Overview on page 1005
- Example: Configuring Persistent NAT with Source NAT Address Pool (CLI) on page 1006
- Example: Configuring Persistent NAT with Interface NAT (CLI) on page 1007

Understanding Persistent NAT

Persistent NAT allows applications to use the Session Traversal Utilities for NAT (STUN) protocol when passing through NAT firewalls (see “Understanding Session Traversal Utilities for NAT (STUN) Protocol” on page 1004). Persistent NAT ensures that all requests from the same internal transport address are mapped to the same *reflexive transport address* (the public IP address and port created by the NAT device closest to the STUN server).

The following types of persistent NAT can be configured on the Juniper Networks device:

- Any remote host—All requests from a specific internal IP address and port are mapped to the same reflexive transport address. Any external host can send a packet to the internal host by sending the packet to the reflexive transport address.
- Target host—All requests from a specific internal IP address and port are mapped to the same reflexive transport address. An external host can send a packet to an internal host by sending the packet to the reflexive transport address. The internal host must have previously sent a packet to the external host's IP address.
- Target host port—All requests from a specific internal IP address and port are mapped to the same reflexive transport address. An external host can send a packet to an internal host by sending the packet to the reflexive transport address. The internal host must have previously sent a packet to the external host's IP address and port.

You configure any of the persistent NAT types with source NAT rules. The source NAT rule action can use a source NAT pool (with or without port translation) or an egress interface. Persistent NAT is not applicable for destination NAT, because persistent NAT bindings are based on outgoing sessions from internal to external.



NOTE: Port overloading is used in Junos OS only for normal interface NAT traffic. Persistent NAT does not support port overloading, and you must explicitly disable port overloading with the `port-overloading off` option at the `[edit security nat source]` hierarchy level.

To configure security policies to permit or deny persistent NAT traffic, you can use two new predefined services—`junos-stun` and `junos-persistent-nat`.



NOTE: Persistent NAT is different from the persistent address feature (see [Example: Configuring a Persistent Address \(CLI\)](#)). The persistent address feature applies to address mappings for source NAT pools configured on the device. The persistent NAT feature applies to address mappings on an external NAT device, and is configured for a specific source NAT pool or egress interface. Also, persistent NAT is intended for use with STUN client/server applications.

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- [Understanding Session Traversal Utilities for NAT \(STUN\) Protocol on page 1004](#)
- [Persistent NAT Configuration Overview on page 1005](#)
- [Understanding Source NAT on page 960](#)
- [Example: Configuring Persistent NAT with Source NAT Address Pool \(CLI\) on page 1006](#)
- [Example: Configuring Persistent NAT with Interface NAT \(CLI\) on page 1007](#)

Understanding Session Traversal Utilities for NAT (STUN) Protocol

Many video and voice applications do not work properly in a NAT environment. For example, Session Initiation Protocol (SIP), used with VoIP, encodes IP addresses and port numbers within application data. If a NAT firewall exists between the requestor and receiver, the translation of the IP address and port number in the data invalidates the information.

Also, a NAT firewall does not maintain a pinhole for incoming SIP messages. This forces the SIP application to either constantly refresh the pinhole with SIP messages or use an ALG to track registration, a function that may or may not be supported by the gateway device.

The Session Traversal Utilities for NAT (STUN) protocol, first defined in *RFC 3489, Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)* and then later in *RFC 5389, Session Traversal Utilities for NAT*, is a simple client/server protocol. A STUN client sends requests to a STUN server, which returns responses to the client. A STUN client is usually part of an application that requires a public IP address and/or port. STUN clients can reside in an end system such as a PC or in a network server whereas STUN servers are usually attached to the public Internet.



NOTE: Both the STUN client and STUN server must be provided by the application. Juniper Networks does not provide a STUN client or server.

The STUN protocol allows a client to:

- Discover whether the application is behind a NAT firewall.

- Determine the type of NAT binding being used (see “Understanding Persistent NAT” on page 1003).
- Learn the reflexive transport address, which is the IP address and port binding allocated by NAT device closest to the STUN server. (There may be multiple levels of NAT between the STUN client and the STUN server.)

The client application can use the IP address binding information within protocols such as SIP and H.323.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Persistent NAT Configuration Overview on page 1005
 - Understanding Persistent NAT on page 1003

Persistent NAT Configuration Overview

To configure persistent NAT, specify the following options with the source NAT rule action (for either a source NAT pool or an egress interface):

- The type of persistent NAT—One of the following: any remote host, target host, or target host port (see “Understanding Persistent NAT” on page 1003).
- (Optional) Address mapping—This option allows requests from a specific internal IP address to be mapped to the same reflexive IP address; internal and reflexive ports can be any ports. An external host using any port can send a packet to the internal host by sending the packet to the reflexive IP address (with a configured incoming policy that allows external to internal traffic). If this option is not configured, the persistent NAT binding is for specific internal and reflexive transport addresses.

You can only specify the **address-mapping** option when the persistent NAT type is any remote host and the source NAT rule action is one of the following actions:

- Source NAT pool with IP address shifting
- Source NAT pool with no port translation and no overflow pool
- (Optional) Inactivity timeout—Time, in seconds, that the persistent NAT binding remains in the device’s memory when all the sessions of the binding entry have expired. When the configured timeout is reached, the binding is removed from memory. The default value is 300 seconds. Configure a value from 60 through 7200 seconds.

When all sessions of a persistent NAT binding have expired, the binding remains in a query state in the SRX Series device’s memory for the specified inactivity timeout period. The query binding is automatically removed from memory when the inactivity timeout period expires (the default is 300 seconds). You can explicitly remove all or specific persistent NAT query bindings with the **clear security nat source persistent-nat-table** command.

- (Optional) Maximum session number—Maximum number of sessions with which a persistent NAT binding can be associated. The default is 30 sessions. Configure a value from 8 through 100.

For interface NAT, you need to explicitly disable port overloading with the **port-overloading off** option at the `[edit security nat source]` hierarchy level.

Finally, there are two predefined services that you can use in security policies to permit or deny STUN and persistent NAT traffic:

- **junos-stun**—STUN protocol traffic.
- **junos-persistent-nat**—Persistent NAT traffic.

For the **any remote host** persistent NAT type, the direction of the security policy is from external to internal. For target host or target host port persistent NAT types, the direction of the security policy is from internal to external.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Example: Configuring Persistent NAT with Source NAT Address Pool (CLI) on page 1006
 - Example: Configuring Persistent NAT with Interface NAT (CLI) on page 1007
 - Understanding Persistent NAT on page 1003
 - Understanding Session Traversal Utilities for NAT (STUN) Protocol on page 1004

Example: Configuring Persistent NAT with Source NAT Address Pool (CLI)

You can configure any of the persistent NAT types with source NAT rules. The example in this section shows how to configure persistent NAT when source NAT is performed with a user-defined address pool.

The following example configures the target host persistent NAT type when source NAT is performed. In the following configuration, the source NAT address pool **sp1** consists of the address 30.1.1.5/32. The source NAT rule set **srs1** configures the following:

- Traffic direction is from zone **internal** to zone **external**.
- For packets with source address in the 40.1.1.0/24 subnet (internal phones) and destination address 20.20.20.0/24 (including STUN server, SIP proxy server and external phones), use the source NAT pool **sp1** to perform source NAT with the target host persistent NAT type.
- Set the persistent NAT **inactivity-timeout** to 180 seconds.

To configure the source NAT address pool:

```
user@host# set security nat source pool sp1 address 30.1.1.5/32
```

To configure the source NAT rule set:

```
user@host# set security nat source rule-set srs1 from zone internal
user@host# set security nat source rule-set srs1 to zone external
user@host# set security nat source rule-set srs1 rule sr1 match source-address 40.1.1.0/24
user@host# set security nat source rule-set srs1 rule sr1 match destination-address 20.20.20.0/24
user@host# set security nat source rule-set srs1 rule sr1 then source-nat pool sp1
user@host# set security nat source rule-set srs1 rule sr1 then source-nat pool persistent-nat permit target-host
```

```
user@host# set security nat source rule-set srs1 rule sr1 then source-nat pool persistent-nat
inactivity-timeout 180
```

For the target host persistent NAT type, configure a security policy to allow persistent NAT traffic from the internal network (internal zone) to the external network (external zone).

To configure a security policy to allow STUN traffic from internal SIP phones to an external STUN server:

```
user@host# set security policies from-zone internal to-zone external policy stun_traffic
match source-address internal_phones destination-address stun_server application
junos-stun
user@host# set security policies from-zone internal to-zone external policy stun_traffic
then permit
```

To configure a security policy to allow SIP proxy traffic from internal SIP phones to an external SIP proxy server:

```
user@host# set security policies from-zone internal to-zone external policy
sip_proxy_traffic match source-address internal_phones destination-address
sip_proxy_server application junos-sip
user@host# set security policies from-zone internal to-zone external policy
stun_proxy_traffic then permit
```

To configure a security policy to allow SIP traffic from internal to external SIP phones:

```
user@host# set security policies from-zone internal to-zone external policy sip_traffic
match source-address internal_phones destination-address external_phones application
junos-persistent-nat
user@host# set security policies from-zone internal to-zone external policy sip_traffic
then permit
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Persistent NAT on page 1003
 - Persistent NAT Configuration Overview on page 1005

Example: Configuring Persistent NAT with Interface NAT (CLI)

You can configure any of the persistent NAT types with source NAT rules. The example in this section shows how to configure persistent NAT when interface NAT is used to perform source NAT. For interface NAT, port overloading must be disabled.

The following example configures the **any remote host** persistent NAT type when interface NAT is performed. The interface NAT rule set **int1** configures the following:

- Traffic direction is from interface **ge-0/0/1.0** to interface **ge-0/0/2.0**.
- For packets with source address 40.1.1.0/24 (internal phones) and destination address 20.20.20.0/24 (including STUN server, SIP proxy server and external phones), perform interface NAT with the **any remote host** persistent NAT type.

You must also disable port overloading for interface NAT.

To configure the interface NAT rule set:

```
user@host# set security nat source rule-set int1 from interface ge-0/0/1.0
user@host# set security nat source rule-set int1 to interface ge-0/0/2.0
user@host# set security nat source rule-set int1 rule in1 match source-address 40.1.1.0/24
user@host# set security nat source rule-set int1 rule in1 match destination-address
20.20.20.0/24
user@host# set security nat source rule-set int1 rule in1 then source-nat interface
persistent-nat permit any-remote-host
```

To disable port overloading for interface NAT:

```
user@host# set security nat source interface port-overloading off
```

For the any remote host persistent NAT type, configure a security policy to allow persistent NAT traffic from the external network (external zone) to the internal network (internal zone).

To configure a security policy to allow STUN traffic from the internal SIP phones to the external STUN server:

```
user@host# set security policies from-zone internal to-zone external policy stun_traffic
match source-address internal_phones destination-address stun_server application
junos-stun
user@host# set security policies from-zone internal to-zone external policy stun_traffic
then permit
```

To configure a security policy to allow SIP proxy traffic from the internal SIP phones to the external SIP proxy server:

```
user@host# set security policies from-zone internal to-zone external policy
sip_proxy_traffic match source-address internal_phones destination-address
sip_proxy_server application junos-sip
user@host# set security policies from-zone internal to-zone external policy
stun_proxy_traffic then permit
```

To configure a security policy to allow SIP traffic from external SIP phones to internal SIP phones:

```
user@host# set security policies from-zone external to-zone internal policy sip_traffic
match source-address external_phones destination-address internal_phones application
junos-persistent-nat
user@host# set security policies from-zone external to-zone internal policy sip_traffic
then permit
```

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Understanding Persistent NAT on page 1003
 - Persistent NAT Configuration Overview on page 1005

Configuring Proxy ARP (CLI Procedure)

You use NAT proxy ARP functionality to configure proxy ARP entries for IP addresses that require either source or destination NAT and that are in the same subnet as the ingress interface.



NOTE: On SRX Series devices, you must explicitly configure NAT proxy ARP.

When configuring NAT proxy ARP, you must specify the logical interface on which to configure proxy ARP. Then you enter an address or address range.

The device performs proxy ARP for the following conditions:

- When addresses defined in the static NAT and source NAT pool are in the same subnet as that of the ingress interface
- When addresses in the original destination address entry in the destination NAT rules are in the same subnet as that of the ingress interface

```
user@host# set security nat proxy-arp interface fe-0/0/0.0 address 10.1.1.10 to 10.1.1.20
```

Related Topics

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Static NAT Configuration Overview on page 932
- Destination NAT Configuration Overview on page 944
- Source NAT Configuration Overview on page 965

Verifying NAT Configuration

Purpose The NAT trace options hierarchy configures trace file and flags for verification purposes. J Series and SRX Series devices have two main components. Those are the Routing Engine (RE) and the Packet Forwarding Engine (PFE). The PFE is divided into the ukernel portion and the real-time portion. For verification, you can turn on flags individually to debug NAT functionality on the RE, ukernel PFE, or real-time PFE. The trace data is written to `/var/log/security-trace` by default.



NOTE: If session logging has been enabled in the policy configurations on the device, the session logs will include specific NAT details for each session. See “Monitoring Policy Statistics” on page 132 for information on how to enable session logging and “Information Provided in Session Log Entries for SRX Series Services Gateways” on page 15 for a description of information provided in session logs.

Use the `security nat traceoptions` command to verify if the NAT configurations are correctly updated to the device upon commit. To verify if NAT translations are being applied to the traffic and to view individual traffic flow processing with NAT translations, use the `security flow traceoptions` command.

Action

```
user@host# set security nat traceoptions flag all
user@host# set security nat traceoptions flag destination-nat-pfe
user@host# set security nat traceoptions flag destination-nat-re
user@host# set security nat traceoptions flag destination-nat-rti
user@host# set security nat traceoptions flag destination-nat-pfe
user@host# set security nat traceoptions flag source-nat-pfe
```

```
user@host# set security nat traceoptions flag source-nat-re
user@host# set security nat traceoptions flag source-nat-rt
user@host# set security nat traceoptions flag static-nat-pfe
user@host# set security nat traceoptions flag static-nat-re
user@host# set security nat traceoptions flag static-nat-rt
```

To filter a specific flow, you can define a packet filter and use it as a traceoption :

```
root@host# set security flow traceoptions packet-filter packet-filter
root@host# set security flow traceoptions packet-filter packet-filter apply-groups
root@host# set security flow traceoptions packet-filter packet-filter apply-groups-except
root@host# set security flow traceoptions packet-filter packet-filter destination-port
root@host# set security flow traceoptions packet-filter packet-filter destination-prefix
root@host# set security flow traceoptions packet-filter packet-filter interface
root@host# set security flow traceoptions packet-filter packet-filter protocol
root@host# set security flow traceoptions packet-filter packet-filter source-port
root@host# set security flow traceoptions packet-filter packet-filter source-prefix
```

To verify NAT traffic and to enable all traffic trace in data plane, use the traceoption **set security flow traceoptions flag basic-datapath** command.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - Static NAT Configuration Overview on page 932
 - Destination NAT Configuration Overview on page 944
 - Source NAT Configuration Overview on page 965

PART 13

GPRS

- General Packet Radio Service on page 1013

CHAPTER 43

General Packet Radio Service

- GPRS Overview on page 1013
- Policy-Based GTP on page 1016
- GTP Inspection Objects on page 1018
- GTP Message Filtering on page 1018
- GTP Information Elements on page 1024
- Understanding GGSN Redirection on page 1030

GPRS Overview

General Packet Radio Service (GPRS) networks connect to several external networks including those of roaming partners, corporate customers, GPRS Roaming Exchange (GRX) providers, and the public Internet. GPRS network operators face the challenge of protecting their network while providing and controlling access to and from these external networks. Juniper Networks provides solutions to many of the security problems plaguing GPRS network operators.

In the GPRS architecture, the fundamental cause of security threats to an operator's network is the inherent lack of security in the GPRS tunneling protocol (GTP). GTP is the protocol used between GPRS support nodes (GSNs). GTP is used to establish a GTP tunnel for individual mobile stations (MSs) and between a Serving GPRS Support Node (SGSN) and a gateway GPRS support node (GGSN). A GTP tunnel is a channel between GSNs through which two hosts exchange data. The SGSN receives packets from the MS and encapsulates them within a GTP header before forwarding them to the GGSN through the GTP tunnel. When the GGSN receives the packets, it decapsulates them and forwards them to the external host.

Communication between different GPRS networks is not secure because GTP does not provide any authentication, data integrity, or confidentiality protection. Implementing IP Security (IPsec) for connections between roaming partners, setting traffic rate limits, and using stateful inspection can eliminate a majority of the GTP's security risks. The GTP firewall features in Junos OS address key security issues in mobile operators' networks.

Juniper Networks security devices mitigate a wide variety of attacks on the following types of GPRS interfaces:

- Gn—The Gn interface is the connection between an SGSN and a GGSN within the same public land mobile network (PLMN).
- Gp—The Gp interface is the connection between two PLMNs.
- Gi—The Gi interface is the connection between a GGSN and the Internet or destination networks connected to a PLMN.



NOTE: The term *interface* has different meanings in Junos OS and in GPRS technology. In Junos OS, an interface is a doorway to a security zone that allows traffic to enter and exit the zone. In GPRS, an interface is a connection, or a reference point, between two components of a GPRS infrastructure, for example, an SGSN and a GGSN.

This topic contains the following sections:

- Gp and Gn Interfaces on page 1014
- Gi Interface on page 1015
- Operational Modes on page 1015

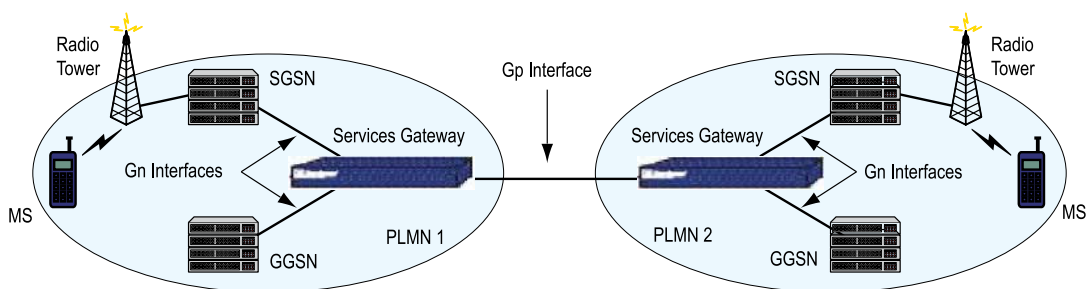
Gp and Gn Interfaces

You implement a security device on the Gn interface to protect core network assets such as the SGSN and GGSN. To secure GTP tunnels on the Gn interface, you place the security device between SGSNs and GGSNs within a common PLMN.

When you implement a security device to the Gp interface, you protect a PLMN from another PLMN. To secure GTP tunnels on the Gp interface, you place the SGSNs and GGSNs of a PLMN behind the security device so that all traffic, incoming and outgoing, goes through the firewall.

Figure 121 on page 1014 illustrates the placement of Juniper Networks SRX Series devices used to protect PLMNs on the Gp and Gn interfaces

Figure 121: Gp and Gn Interfaces



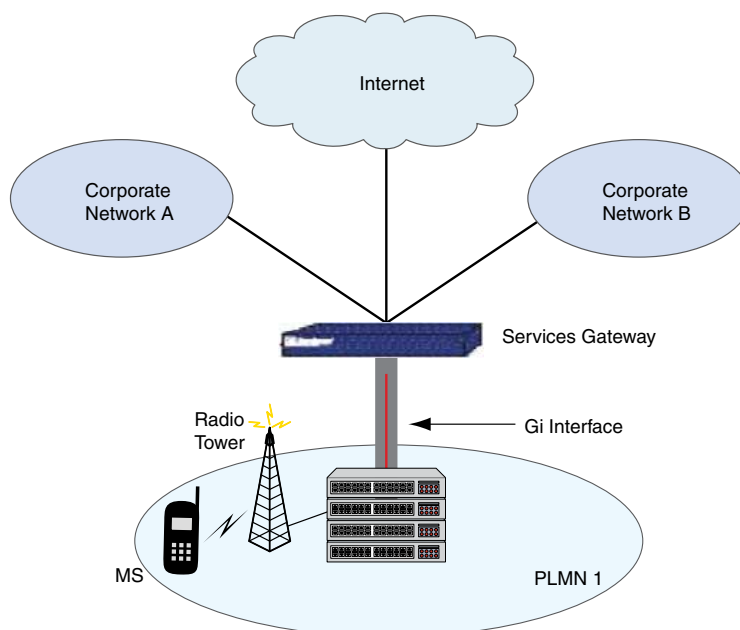
Gi Interface

When you implement a security device on the Gi interface, you can simultaneously control traffic for multiple networks, protect a PLMN against the Internet and external networks, and protect mobile users from the Internet and other networks. Junos OS provides a great number of virtual routers, making it possible for you to use one virtual router per customer network and thereby allow the separation of traffic for each customer network.

The security device can securely forward packets to the Internet or destination networks using the Layer 2 Tunneling Protocol (L2TP) for IPsec virtual private network (VPN) tunnels. (Note, however, that SRX Series devices do not support full L2TP.)

Figure 122 on page 1015 illustrates the implementation of a security device to protect a PLMN on the Gi interface.

Figure 122: Gi Interface



Operational Modes

Junos OS supports two interface operational modes with GTP: transparent mode and route mode. If you want the security device to participate in the routing infrastructure of your network, you can run it in route mode. This requires a certain amount of network redesign. Alternatively, you can implement the security device into your existing network in transparent mode without having to reconfigure the entire network. In transparent mode, the security device functions as a Layer 2 switch or bridge, and the IP addresses of interfaces are set at 0.0.0.0, making the presence of the security device invisible, or *transparent*, to users.

Junos OS supports Network Address Translation (NAT) on interfaces and policies that do not have GTP inspection enabled.

Currently in Junos OS, route mode supports active/passive, and active/active chassis cluster. Transparent mode supports active/passive only.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - *Junos OS Layer 2 Bridging and Switching Configuration Guide for Security Devices*
 - Chassis Cluster Overview on page 795

Policy-Based GTP

- Understanding Policy-Based GTP on page 1016
- Example: Enabling GTP Inspection in Policies (CLI) on page 1017

Understanding Policy-Based GTP

By default, the public land mobile network (PLMN) that the Juniper Networks device protects is in the Trust zone. The device protects the PLMN in the Trust zone against other PLMNs in other zones. You can place all the PLMNs against which you are protecting your PLMN in the Untrust zone, or you can create user-defined zones for each PLMN. A PLMN can occupy one security zone or multiple security zones.

You must create policies to enable traffic to flow between zones and PLMNs. Policies contain rules that permit, deny, or tunnel traffic. The device performs GPRS tunneling protocol (GTP) policy filtering by checking every GTP packet against policies that regulate GTP traffic and by then forwarding, dropping, or tunneling the packet based on these policies.

By selecting the GTP service in a policy, you enable the device to permit, deny, or tunnel GTP traffic. However, this does not enable the device to inspect GTP traffic. For the device to inspect GTP traffic, you must apply a GTP configuration, also referred to as a *GTP inspection object*, to a policy.

You can apply only one GTP inspection object per policy, but you can apply a GTP inspection object to multiple policies. Using policies, you can permit or deny the establishment of GTP tunnels from certain peers such as a Serving GPRS Support Node (SGSN).

You can configure policies that specify “Any” as the source or destination zone (thereby including all hosts in the zone), and you can configure policies that specify multiple source and destination addresses.

In policies, you can enable traffic logging.

- Related Topics**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Enabling GTP Inspection in Policies (CLI)

In this example, you configure interfaces and create addresses and two policies to allow bidirectional traffic between two networks within the same PLMN. You also apply a GTP inspection object to the policies.

1. Enable GTP:

```
user@host# set security gprs gtp enable
user@host# commit
user@host# exit
user@host# request system reboot
```



NOTE: GTP is disabled by default. The device must be restarted after enabling GTP.

2. Create the GTP inspection object:

```
user@host# set security gprs gtp profile gtp1
```

3. Configure interfaces:

```
user@host# set interfaces ge-0/0/1 unit 0 family inet address 2.0.0.254/8
user@host# set interfaces ge-0/0/2 unit 0 family inet address 3.0.0.254/8
```

4. Configure security zones:

```
user@host# set security zones security-zone sgsn interfaces ge-0/0/1.0
user@host# set security zones security-zone sgsn host-inbound-traffic
  system-services all
user@host# set security zones security-zone sgsn host-inbound-traffic protocols all
user@host# set security zones security-zone ggsn interfaces ge-0/0/2.0
user@host# set security zones security-zone ggsn host-inbound-traffic
  system-services all
user@host# set security zones security-zone ggsn host-inbound-traffic protocols all
```

5. Specify addresses:

```
user@host# set security zones security-zone sgsn address-book address local-sgsn
  2.0.0.5/32
user@host# set security zones security-zone ggsn address-book address remote-ggsn
  3.0.0.6/32
```

6. Enable the GTP service in the security policies:

```
user@host# set security policies from-zone sgsn to-zone ggsn policy sgsn_to_ggsn
  match source-address local-sgsn destination-address remote-ggsn application
  junos-gprs-gtp
user@host# set security policies from-zone sgsn to-zone ggsn policy sgsn_to_ggsn
  then permit application-services gprs-gtp-profile gtp1
user@host# set security policies from-zone ggsn to-zone sgsn policy ggsn_to_sgsn
  match source-address remote-ggsn destination-address local-sgsn application
  junos-gprs-gtp
user@host# set security policies from-zone ggsn to-zone sgsn policy sgsn_to_ggsn
  then permit application-services gprs-gtp-profile gtp1
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

GTP Inspection Objects

- Understanding GTP Inspection Objects on page 1018
- Example: Creating a GTP Inspection Object (CLI) on page 1018

Understanding GTP Inspection Objects

For the device to perform the inspection of GPRS tunneling protocol (GTP) traffic, you must create a GTP inspection object and then apply it to a policy. GTP inspection objects provide more flexibility in that they allow you to configure multiple policies that enforce different GTP configurations. You can configure the device to control GTP traffic differently based on source and destination zones and addresses, action, and so on.

To configure GTP features, you must enter the context of a GTP configuration. To save your settings in the CLI, you must first exit the GTP configuration, then enter the **commit** command.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Creating a GTP Inspection Object (CLI)

In this example, you create a GTP inspection object named LA-NY. You preserve most of the default values, but you enable the sequence number validation feature.

```
user@host# set security gprs gtp profile la-ny
user@host# set security gprs gtp profile la-ny seq-number-validated
user@host# commit
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

GTP Message Filtering

- Understanding GTP Message Filtering on page 1018
- GTP Message-Length Filtering on page 1019
- GTP Message-Type Filtering on page 1019
- GTP Message-Rate Limiting on page 1022
- GTP Sequence Number Validation on page 1023
- Understanding GTP IP Fragmentation on page 1023

Understanding GTP Message Filtering

When the device receives a GPRS tunneling protocol (GTP) packet, it checks the packet against policies configured on the device. If the packet matches a policy, the device inspects the packet according to the GTP configuration applied to the policy. If the packet fails to meet any of the GTP configuration parameters, the device will pass or drop the packets based on the configuration of the GTP inspection object.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

GTP Message-Length Filtering

- Understanding GTP Message-Length Filtering on page 1019
- Example: Setting GTP Message Lengths (CLI) on page 1019

Understanding GTP Message-Length Filtering

You can configure the device to drop packets that do not meet your specified minimum or maximum message lengths. In the GPRS tunneling protocol (GTP) header, the message length field indicates the length, in octets, of the GTP payload. It does not include the length of the GTP header itself, the UDP header, or the IP header. The default minimum and maximum GTP message lengths are 0 and 65,535 bytes, respectively.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Setting GTP Message Lengths (CLI)

In this example, you configure the minimum GTP message length to be 8 octets and the maximum GTP message length to be 1200 octets for the GTP inspection object.

```
user@host# set security gprs gtp profile gtp1
user@host# set security gprs gtp profile gtp1 min-message-length 8
user@host# set security gprs gtp profile gtp1 max-message-length 1200
user@host# commit
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

GTP Message-Type Filtering

- Understanding GTP Message-Type Filtering on page 1019
- Example: Permitting and Denying GTP Message Types (CLI) on page 1019
- Supported GTP Message Types on page 1020

Understanding GTP Message-Type Filtering

You can configure the device to filter GPRS tunneling protocol (GTP) packets and permit or deny them based on their message type. By default, the device permits all GTP message types.

A GTP message type includes one or many messages. When you permit or deny a message type, you automatically permit or deny all messages of the specified type. For example, if you select to drop the `sgsn-context` message type, you thereby drop `sgsn-context-request`, `sgsn-context-response`, and `sgsn-context-acknowledge` messages.

You permit and deny message types based on the GTP version number. For example, you can deny message types for one version while you permit them for the other version.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Permitting and Denying GTP Message Types (CLI)

In this example, for the `gtp1` profile, you configure the device to drop the `error-indication` and `failure-report` message types for version 1.

```

user@host# set security gprs gtp profile gtp1
user@host# set security gprs gtp profile gtp1 drop error-indication 1
user@host# set security gprs gtp profile gtp1 drop failure-report 1
user@host# commit

```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Supported GTP Message Types

Table 85 on page 1020 lists the GTP messages supported in GTP Releases 1997 and 1999 (including charging messages for GTP) and the message types that you can use to configure GTP message-type filtering.

Table 85: GTP Messages

Message	Message Type	Version 0	Version 1
create AA pdp context request	create-aa-pdp	b	
create AA pdp context response	create-aa-pdp	b	
create pdp context request	create-pdp	b	b
create pdp context response	create-pdp	b	b
data record request	data-record	b	b
data record response	data-record	b	b
delete AA pdp context request	delete-aa-pdp	b	
delete AA pdp context response	delete-aa-pdp	b	
delete pdp context request	delete-pdp	b	b
delete pdp context response	delete-pdp	b	b
echo request	echo	b	b
echo response	echo	b	b
error indication	error-indication	b	b
failure report request	failure-report	b	b
failure report response	failure-report	b	b
forward relocation request	fwd-relocation	b	b
forward relocation response	fwd-relocation	b	b
forward relocation complete	fwd-relocation	b	b

Table 85: GTP Messages (*continued*)

Message	Message Type	Version 0	Version 1
forward relocation complete acknowledge	fwd-relocation	b	b
forward SRNS context	fwd-srns-context	b	b
forward SRNS context acknowledge	fwd-srns-context	b	b
identification request	identification	b	b
identification response	identification	b	b
node alive request	node-alive	b	b
node alive response	node-alive	b	b
note MS GPRS present request	note-ms-present	b	b
note MS GPRS present response	note-ms-present	b	b
pdu notification request	pdu-notification	b	b
pdu notification response	pdu-notification	b	b
pdu notification reject request	pdu-notification	b	b
pdu notification reject response	pdu-notification	b	b
RAN info relay	ran-info	b	b
redirection request	redirection	b	b
redirection response	redirection	b	b
relocation cancel request	relocation-cancel	b	b
relocation cancel response	relocation-cancel	b	b
send route info request	send-route	b	b
send route info response	send-route	b	b
sgsn context request	sgsn-context	b	b
sgsn context response	sgsn-context	b	b
sgsn context acknowledge	sgsn-context	b	b

Table 85: GTP Messages (*continued*)

Message	Message Type	Version 0	Version 1
supported extension headers notification	supported-extension	b	b
g-pdu	gtp-pdu	b	b
update pdp context request	update-pdp	b	b
updated pdp context response	update-pdp	b	b
version not supported	version-not-supported	b	b

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

GTP Message-Rate Limiting

- Understanding GTP Message-Rate Limiting on page 1022
- Example: Limiting the GTP Message Rate (CLI) on page 1022

Understanding GTP Message-Rate Limiting

You can configure the device to limit the rate of network traffic going to a GPRS support node (GSN). You can set separate thresholds, in packets per second, for GGSN tunneling protocol, control (GTP-C) messages. Because GTP-C messages require processing and replies, they can potentially overwhelm a GSN. By setting a rate limit on GTP-C messages, you can protect your GSNs from possible denial-of-service (DoS) attacks such as the following:

- Border gateway bandwidth saturation—A malicious operator connected to the same GPRS Roaming Exchange (GRX) as your public land mobile network (PLMN) can direct so much network traffic at your Border Gateway that legitimate traffic is starved for bandwidth in or out of your PLMN, thus denying roaming access to or from your network.
- GTP flood—GPRS tunneling protocol (GTP) traffic can flood a GSN, forcing it to spend its CPU cycles processing illegitimate data. This can prevent subscribers from roaming and forwarding data to external networks, and it can prevent a General Packet Radio Service (GPRS) from attaching to the network.

This feature limits the rate of traffic sent to each GSN from the Juniper Networks device. The default rate is unlimited.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Limiting the GTP Message Rate (CLI)

In the following example, you limit the rate of incoming GTP-C messages to 300 packets per second.

```
user@host# set security gprs gtp profile gtp1
user@host# set security gprs gtp profile gtp1 rate-limit 300
```

```
user@host# commit
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

GTP Sequence Number Validation

- Understanding GTP Sequence Number Validation on page 1023
- Example: Enabling GTP Sequence Number Validation (CLI) on page 1023

Understanding GTP Sequence Number Validation

You can configure the device to perform sequence-number validation.

The header of a GPRS tunneling protocol (GTP) packet contains a Sequence Number field. This number indicates to the gateway GPRS support node (GGSN) receiving the GTP packets the order of the packets. During the packet data protocol (PDP) context-activation stage, a sending GGSN uses zero (0) as the sequence number for the first G-PDU it sends through a tunnel to another GGSN. The sending GGSN increments the sequence number for each following G-PDU it sends. The value resets to zero when it reaches 65,535.

During the PDP context-activation stage, the receiving GGSN sets its counter to zero. Subsequently, whenever the receiving GGSN receives a valid G-PDU, the GGSN increments its counter by one. The counter resets to zero when it reaches 65,535.

Normally, the receiving GGSN compares the sequence number in the packets it received with the sequence number from its counter. If the numbers correspond, the GGSN forwards the packet. If they differ, the GGSN drops the packet. By implementing a Juniper Networks device between the GGSNs, the device can perform this validation for the GGSN and drop packets that arrive out of sequence. This feature helps conserve GGSN resources by preventing the unnecessary processing of invalid packets.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Enabling GTP Sequence Number Validation (CLI)

In this example, you enable the sequence number validation feature.

```
user@host# set security gprs gtp profile gtp1
user@host# set security gprs gtp profile gtp1 seq-number-validated
user@host# commit
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Understanding GTP IP Fragmentation

A GPRS tunneling protocol (GTP) packet consists of the message body and three headers: GTP, UDP, and IP. If the resulting IP packet is larger than the maximum transmission unit (MTU) on the transferring link, the sending Serving GPRS Support Node (SGSN) or gateway GPRS support node (GGSN) performs an IP fragmentation.

By default, the device buffers IP fragments until it receives a complete GTP message, and then inspects the GTP message.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

GTP Information Elements

- Understanding GTP Information Elements on page 1024
- GTP APN Filtering on page 1024
- GTP IMSI Prefix Filtering on page 1025
- GTP R6 Information Elements on page 1026

Understanding GTP Information Elements

Information elements (IEs) are included in all GPRS tunneling protocol (GTP) control message packets. IEs provide information about GTP tunnels, such as creation, modification, deletion, and status. Junos OS supports IEs consistent with Third-Generation Partnership Project (3GPP) Release 6. If you have contractual agreements with operators running earlier releases of 3GPP, you can reduce network overhead by restricting control messages containing unsupported IEs.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

GTP APN Filtering

- Understanding GTP APN Filtering on page 1024
- Example: Setting a GTP APN and a Selection Mode (CLI) on page 1025

Understanding GTP APN Filtering

An access point name (APN) is an information element (IE) included in the header of a GPRS tunneling protocol (GTP) packet that provides information about how to reach a network. An APN comprises two elements:

- Network ID—Identifies the name of an external network such as `mobiphone.com`.
- Operator ID—Uniquely identifies the operators' public land mobile network (PLMN) such as `mnc123.mcc456`.

By default, the device permits all APNs. However, you can configure the device to perform APN filtering to restrict access to roaming subscribers to external networks.

To enable APN filtering, you must specify one or more APNs. To specify an APN, you need to know the domain name of the network (for example, `mobiphone.com`) and, optionally, the operator ID. Because the domain name (network ID) portion of an APN can potentially be very long and contain many characters, you can use the wildcard (*) as the first character of the APN. The wildcard indicates that the APN is not limited only to `mobiphone.com` but also includes all the characters that might precede it.

You may also set a *selection mode* for the APN. The selection mode indicates the origin of the APN and whether or not the Home Location Register (HLR) has verified the user subscription. You set the selection mode according to the security needs of your network. Possible selection modes include the following:

- Mobile Station—Mobile station-provided APN, subscription not verified.

This selection mode indicates that the mobile station (MS) provided the APN and that the HLR did not verify the user's subscription to the network.

- Network—Network-provided APN, subscription not verified.

This selection mode indicates that the network provided a default APN because the MS did not specify one, and that the HLR did not verify the user's subscription to the network.

- Verified—MS or network-provided APN, subscription verified.

This selection mode indicates that the MS or the network provided the APN and that the HLR verified the user's subscription to the network.

APN filtering applies only to create-pdp-request messages. When performing APN filtering, the device inspects GTP packets to look for APNs that match APNs that you set. If the APN of a GTP packet matches an APN that you specified, the device then verifies the selection mode and only forwards the GTP packet if both the APN and the selection mode match the APN and the selection mode that you specified. Because APN filtering is based on perfect matches, using the wildcard (*) when setting an APN suffix can prevent the inadvertent exclusion of APNs that you would otherwise authorize. The device automatically denies all other APNs that do not match.

Additionally, the device can filter GTP packets based on the combination of an International Mobile Subscriber Identity (IMSI) prefix and an APN.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Setting a GTP APN and a Selection Mode (CLI)

In this example, you set **mobiphone.com.mnc123.mcc456.gprs** as an APN and use the wildcard (*). You also set **Network** as the selection mode.

```
user@host# set security gprs gtp profile gtp1
user@host# set security gprs gtp profile gtp1 apn *mobiphone.com.mnc123.mcc456.gprs
      mcc-mnc * action selection net
user@host# commit
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

GTP IMSI Prefix Filtering

- Understanding IMSI Prefix Filtering of GTP Packets on page 1025
- Example: Setting a Combined IMSI Prefix and APN Filter (CLI) on page 1026

Understanding IMSI Prefix Filtering of GTP Packets

A GPRS support node (GSN) identifies a mobile station (MS) by its International Mobile Station Identity (IMSI). An IMSI consists of three elements: the mobile country code (MCC), the mobile network code (MNC), and the Mobile Subscriber Identification Number (MSIN). The MCC and MNC combined constitute the IMSI prefix and identify the mobile subscriber's home network, or public land mobile network (PLMN).

By setting IMSI prefixes, you can configure the device to deny GPRS tunneling protocol (GTP) traffic coming from nonroaming partners. By default, a device does not perform IMSI prefix filtering on GTP packets. By setting IMSI prefixes, you configure the device to filter create-pdp-request messages and permit only GTP packets with IMSI prefixes that match the ones you set. The device allows GTP packets with IMSI prefixes that do not match any of the IMSI prefixes that you set. To block GTP packets with IMSI prefixes that do not match any of the IMSI prefixes set, use an explicit wildcard for the IMSI filter, and the drop action should be the last IMSI prefix filtering policy.

When you filter GTP packets based on an IMSI prefix, you must also specify an APN.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Setting a Combined IMSI Prefix and APN Filter (CLI)

In this example, you set `mobiphone.com.mnc123.mcc456.gprs` as an APN and use the wildcard (*). You permit all selection modes for this APN. You also set the IMSI prefix for a known PLMN, which is 246565. The MCC-MNC pair can be five or six digits.

```
user@host# set security gprs gtp profile gtp1
user@host# set security gprs gtp profile gtp1 apn *mobiphone.com.mnc123.mcc456.gprs
      mcc-mnc 246565 action pass
user@host# commit
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

GTP R6 Information Elements

- Understanding R6 Information Elements Removal on page 1026
- Example: Removing R6 Information Elements from GTP Messages (CLI) on page 1026
- Supported R6 Information Elements on page 1027

Understanding R6 Information Elements Removal

The Third-Generation Partnership Project (3GPP) R6 information element (IE) removal feature allows you to retain interoperability in roaming between Second-Generation Partnership Project (2GPP) and 3GPP networks. You can configure the GPRS tunneling protocol (GTP)-aware Juniper Networks device, residing on the border of a public land mobile network (PLMN) and a GPRS Roaming Exchange (GRX) and acting as a Gp firewall, to remove 3GPP-specific attributes from the GTP packet header when the packet passes into a 2GPP network. You can configure the device to remove the RAT, RAI, ULI, IMEI-SV, and access point name (APN) restriction IEs from GTP messages prior to forwarding these messages to the gateway GPRS support node (GGSN).

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Example: Removing R6 Information Elements from GTP Messages (CLI)

In this example, you configure the Gp interface of the security device to remove newly added R6 IEs (RAT, ULI, IMEI-SV and APN restriction) from the GTP message.

```
user@host# set security gprs gtp profile gtp1
user@host# set security gprs gtp profile gtp1 remove-r6
```

```
user@host# commit
save
```

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Supported R6 Information Elements

Junos OS supports all 3GPP R6 IEs for GTP), as listed in Table 86 on page 1027.

Table 86: Supported Information Elements

IE Type Value	Information Element
1	Cause
2	International Mobile Subscriber Identity (IMSI)
3	Routing Area Identity (RAI)
4	Temporary Logical Link Identity (TLLI)
5	Packet TMSI (P-TMSI)
8	Reordering Required
9	Authentication Triplet
11	MAP Cause
12	P-TMSI Signature
13	MS Validated
14	Recovery
15	Selection Mode
16	Tunnel Endpoint Identifier Data I
17	Tunnel Endpoint Identifier Control Plane
18	Tunnel Endpoint Identifier Data II
19	Teardown ID
20	NSAPI
21	RANAP Cause
22	RAB Context
23	Radio Priority SMS

Table 86: Supported Information Elements (*continued*)

IE Type Value	Information Element
24	Radio Priority
25	Packet Flow ID
26	Charging Characteristics
27	Trace Reference
28	Trace Type
29	MS Not Reachable Reason
127	Charging ID
128	End User Address
129	MM Context
130	PDP Context
131	Access Point Name
132	Protocol Configuration Options
133	GSN Address
134	MS International PSTN/ISDN Number (MSISDN)
135	Quality of Service Profile
136	Authentication Quintuplet
137	Traffic Flow Template
138	Target Identification
139	UTRAN Transparent Container
140	RAB Setup Information
141	Extension Header Type List
142	Trigger Id
143	OMC Identity
144	RAN Transparent Container

Table 86: Supported Information Elements (*continued*)

IE Type Value	Information Element
145	PDP Context Prioritization
146	Additional RAB Setup Information
147	SGSN Number
148	Common Flags
149	APN Restriction
150	Radio Priority LCS
151	RAT Type
152	User Location Information
153	MS Time Zone
154	IMEI-SV
155	CAMEL Charging Information Container
156	MBMS UE Context
157	Temporary Mobile Group Identity (TMGI)
158	RIM Routing Address
159	MBMS Protocol Configuration Options
160	MBMS Service Area
161	Source TNC PDCP context Information
162	Additional Trace Information
163	Hop Counter
164	Selected PLMN ID
165	MBMS Session Identifier
166	MBMS2G/3G Indicator
167	Enhanced NSAPI
168	MBMS Session Duration

Table 86: Supported Information Elements *(continued)*

IE Type Value	Information Element
169	Additional MBMS Trace Information
251	Charging Gateway Address
255	Private Extension

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

Understanding GGSN Redirection

Junos OS supports GPRS tunneling protocol (GTP) traffic and gateway GPRS support node (GGSN) redirection. A GGSN (X) can send create-pdp-context responses in which it can specify different GGSN IP addresses (GGSN Y and GGSN Z) for subsequent GTP-C and GTP-U messages. Consequently, the SGSN sends the subsequent GGSN tunneling protocol, control (GTP-C) and GGSN tunneling protocol, user plane (GTP-U) messages to GGSNs Y and Z, instead of X.

Related Topics *Junos OS Feature Support Reference for SRX Series and J Series Devices*

PART 14

Index

- Index on page 1033

Index

Symbols

#, comments in configuration statements.....xli	
(), in syntax descriptions.....xli	
3DES.....359	
< >, in syntax descriptions.....xli	
[], in configuration statements.....xli	
{ }, in configuration statements.....xli	
(pipe), in syntax descriptions.....xli	
.....208, 720, 743	
<Emphasis> See <Default Para Font> DoS	
attributes.....715	
network.....748	
OS-specific.....761	
SDP.....208	

A

AAA.....312	
Access Manager	
adding firewall connection to.....413	
auto-upgrading.....412	
client-side files.....415	
downloading to user's computers.....411	
error messages.....418	
launching from client.....413	
logging.....418	
overview.....405	
system requirements.....414	
Windows registry changes.....417	
Access Point Name See APN	
access profile	
configuring	
dynamic VPN.....407	
accommodating end-to-end TCP communication	
end-to-end TCP communication.....77	
address sweep.....711	
Advanced Encryption Standard (AES).....359	
AES.....359	
agentless access See UAC, Infranet agent	
agents, zombie.....743	

aggressive mode.....368	
AH (authentication header) protocol	
overview.....355	
ALGs	
MS RPC.....291	
SIP.....207	
SIP NAT.....221	
Sun RPC.....288	
allowing	
unknown SIP ALG message types.....218	
antireplay	
group VPN.....453	
antispam filtering.....587	
local list.....595	
message handling.....603	
server-based.....587, 589, 596	
antivirus	
verifying.....644	
antivirus, express.....649	
EICAR file.....650	
limitations.....650	
testing.....650	
updating antivirus patterns.....651	
antivirus, full.....605	
application protocol scanning.....616	
content size limits.....613	
decompression layer limit.....613	
file extension scanning.....611	
intelligent prescreening.....612	
notification options.....626, 627, 628	
scan session throttling.....615	
scanning timeout.....614	
signature database support.....606	
updating antivirus patterns.....606	
APN	
filtering.....1024	
selection mode.....1024	
appDDoS	
application-level DDoS protection	
overview.....523	

AppDDoS		
understanding logging.....	572	
AppDDoS Protection		
enabling example.....	528	
application binding.....	497, 550	
application identification.....	549	
application binding.....	550	
application package manual download.....	772	
configuring policies (IDP).....	551	
custom application definitions.....	777	
disable.....	552, 775	
memory limit.....	557	
nested applications.....	553, 776	
overview.....	549	
service binding.....	550	
session limit.....	557	
system cache.....	555, 783	
system caching for nested application		
identification.....	555, 784	
understanding application package.....	770	
verifying application package.....	774	
verifying cache statistics.....	556, 784	
verifying counters.....	559	
<i>See also</i> IDP		
application identification (Junos)		
overview.....	769	
application identification services		
memory limit.....	786	
session limit.....	786	
application package		
automatic update.....	773	
manually update.....	772	
understanding.....	770	
updating, overview.....	771	
verifying.....	774	
application sets		
IDP, configuring.....	495	
overview.....	493	
application system cache.....	555, 783	
overview.....	555, 783	
application tracking		
AppTrack.....	789	
application-level DDoS.....	523	
application-level DDoS protection		
overview.....	523	
configuring statistic reporting.....	533	
statistics reporting overview.....	531	
Application-Level DDoS		
understanding logging.....	572	
application-level DDoS protection		
configuration.....	528	
applications		
IDP, configuring.....	494	
AppTrack		
application tracking.....	789	
associating policy to schedulers.....	137	
attack detection		
overview.....	709	
attack object groups.....	540	
predefined.....	540	
attack objects		
predefined.....	540	
attacks		
DOS.....	743, 764	
ICMP		
floods.....	758, 759	
fragments.....	736	
IP packet fragments.....	740	
Land.....	760, 761	
large ICMP packets.....	737	
Ping of Death.....	762	
replay.....	372	
session table floods.....	726, 744	
SYN floods.....	748, 753	
SYN fragments.....	741	
Teardrop.....	763, 764	
UDP floods.....	759, 760	
unknown protocols.....	739	
WinNuke.....	764, 765	
auth users		
groups.....	321	
authenticating users		
pass-through authentication.....	298	
authentication		
administrative.....	312	
algorithms.....	359	
client groups.....	321	
configuring		
external authentication servers.....	314	
SecurID server.....	314	
pass-through.....	298	
Web.....	305	
authentication tables <i>See</i> UAC, authentication		
tables		
authentication, authorization, and accounting		
servers.....	312	
AutoKey IKE VPN.....	357	
management.....	357	

B

banners.....	323
braces, in configuration statements.....	xli
brackets	
angle, in syntax descriptions.....	xli
square, in configuration statements.....	xli

C

CA certificates.....	383
captive portal	
captive portal policy creating.....	346
configuration.....	345
overview.....	344
redirect URL configure.....	349
redirect URL options.....	348
certificates.....	357
CA.....	383
loading.....	395
local.....	388
revocation.....	401
self-signed.....	398
UAC deployments <i>See</i> UAC, device	
authentication	
changing session characteristics.....	7, 73
chassis cluster	
ISSU upgrading.....	874
chassis clusters	
about.....	795
control interfaces.....	856
creating a J Series cluster.....	864
creating an SRX Series cluster.....	861
disabling.....	877
enabling.....	845
fabric interfaces.....	856
formation.....	796
hardware setup for J Series devices.....	863
hardware setup for SRX Series devices.....	857
management interfaces on J Series	
devices.....	855
management interfaces on SRX Series	
devices.....	855
node interfaces on J Series devices.....	853
node interfaces on SRX Series devices.....	846
redundancy groups.....	797
setting node and cluster IDs.....	866
verifying.....	869
verifying interfaces.....	821
verifying redundancy group status.....	804

verifying statistics.....	870
verifying status.....	872
client groups for firewall authentication.....	321
cold sync	
monitoring.....	811
colocation mode.....	444
comments, in configuration statements.....	xli
compiling IDP policy.....	546
compound attack sample.....	511
conditional route advertising configuration.....	825
configuring	
anomaly attack objects.....	517
application identification services, memory	
limit.....	786
application identification services, session	
limit.....	786
application identification, memory limit.....	557
application identification, session limit.....	557
AutoKey IKE.....	374
chassis cluster information.....	869
conditional route advertising.....	825
control link recovery.....	837
control ports.....	830
dampening time between back-to-back	
redundancy group failovers.....	816
DSCP in IDP policy.....	490
dynamic VPN client configurations.....	408
dynamic VPN global settings.....	409
exempt rulebase.....	485
external authentication servers.....	314
fabric.....	842
group VPN.....	432
group VPN colocation mode.....	445
group VPN multicast rekey.....	456
group VPN unicast rekey.....	454
group VPNs.....	431
host inbound traffic.....	90
protocols.....	95
IDP application sets.....	495
IDP applications.....	494
IDP in security policy.....	465
IDP policy, application identification.....	551
IDP services.....	494
IKE gateway and peer authentication.....	370
IKE policy, authentication, and proposal.....	370
interface monitoring.....	805
interface source NAT for incoming SIP	
calls.....	230

interface source NAT pool for incoming SIP		configuring.....	515, 517
calls.....	232	name.....	496
IPS rulebase.....	481	protocol anomaly.....	508
IPsec policy.....	373	protocol binding.....	500
IPsec tunnel overview.....	366	service binding.....	497
log suppression.....	574	severity.....	496
management interfaces.....	867	signature.....	503
Phase 2 proposals.....	373	time binding.....	502
redundancy groups.....	803	customer support.....	xlii
redundant Ethernet interfaces.....	819	contacting JTAC.....	xlii
SCCP DoS attack protection.....	254, 255		
signature attack objects.....	515	D	
signature database automatic download.....	545	data	
signature database manual download.....	543	fabric.....	840
SIP DoS attack protection.....	216	fabric (dual).....	841
SIP proxy		forwarding.....	839
private zone.....	235	plane.....	838
public zone.....	237	Data Encryption Standard (DES).....	359
static NAT for incoming SIP calls.....	234	data path.....	79
TCP-reset parameter.....	98	fast-path processing.....	82
terminal rules.....	488	forward processing.....	80
three-zone SIP scenario.....	238	session-based processing.....	80
VPN global settings.....	375	data processing, stateful and stateless.....	3, 69
Content Filtering.....	665	DDoS.....	743
filter types.....	665	application-level.....	523
protocol support.....	666	defining	
verifying.....	677	exempt rulebase.....	485
control link.....	829	IPS rulebase.....	481
failure and recovery.....	835	DES.....	359
control link recovery		destination NAT.....	941
configuring.....	837	address and port translation configuration	
control plane		example.....	949
overview.....	828	address pools.....	942
control ports		configuration overview.....	944
configuring.....	830	overview.....	942
controlling session termination.....	74	rules.....	943
conventions		single address translation configuration	
notice icons.....	xl	example.....	944
text and syntax.....	xl	subnet translation configuration	
cookies, SYN.....	756	example.....	955
CoS features.....	6, 73	with source NAT configuration example.....	996
counters, verifying		Diffie-Hellman.....	358
for application identification.....	559	Diffserv	
creating a J Series chassis cluster.....	864	configuring in IDP policy.....	490
creating an SRX Series chassis cluster.....	861	digital signature.....	386
curly braces, in configuration statements.....	xli	disabling	
custom attacks		chassis clusters.....	877
application binding.....	497	disabling TCP packet security checks.....	76
compound.....	509		

- documentation
 - comments on.....xliv
- DoS
 - firewall.....747
 - session table floods.....726, 744
- DoS attacks.....743
- download
 - Access Manager.....411
 - client configuration, dynamic VPN.....411
 - signature database automatic.....545
 - signature database manually.....543
 - signature database overview.....542
- dual control links
 - about.....830
 - connecting.....832
 - upgrading the second routing engine.....833
- dynamic auth table provisioning See UAC, dynamic
- auth table provisioning
- dynamic packet filtering.....709
- dynamic policies See group VPNs
- dynamic VPNs
 - client configurations.....408
 - configuration overview.....407
 - downloading client configurations.....411, 412
 - global settings.....409
 - overview.....405
- E**
 - enabling chassis clusters.....845
 - encryption algorithms.....359
 - ESP.....358, 359
 - ESP (Encapsulating Security Payload) protocol
 - overview.....355
 - exempt rulebase
 - configuring.....485
- F**
 - fabric configuration.....842
 - fabric data link.....840
 - fabric data link (dual).....841
 - fabric data-link failure.....840
 - fabric interfaces.....856
 - fast-path processing.....82
 - filters, stateless firewall.....6, 72
 - FIN scans.....725
 - FIN without ACK flag attack detection
 - overview.....722
 - firewall users, pass-through
 - authentication process.....298
 - floods
 - ICMP.....758, 759
 - session table.....744
 - SYN.....748, 753, 756
 - UDP.....759, 760
 - flow-based packet processing
 - defined.....3
 - flow-based processing
 - enabling.....61
 - flowd
 - monitoring.....810
 - font conventions.....xl
 - forward processing.....80
 - forwarding features.....82
- G**
 - gatekeeper devices.....173
 - GDOI protocol See group VPNs
 - Gi interface.....1013
 - glossary
 - IDP policy.....464
 - Gp interface.....1013
 - gprs
 - about.....1013
 - tunneling protocol.....1013
 - group keys
 - KEK.....428
 - TEK.....428
 - group policies See group VPNs
 - group VPNs
 - antireplay.....453
 - colocation configuration.....445
 - colocation mode.....444
 - configuration.....432
 - configuration overview.....431
 - dynamic policies.....427
 - GDOI protocol.....425
 - group keys.....428
 - group policies.....427
 - heartbeat messages.....457
 - IKE Phase 1 configuration.....450
 - interoperability with GET VPN.....458
 - IPsec SA configuration.....451
 - key activation.....431
 - limitations.....458
 - member.....426
 - member reregistration.....430
 - multicast rekey configuration.....456
 - overview.....423

rekey messages.....	429	detector.....	512
scope policies.....	427	DSCP.....	490
server.....	426	enabling IDP.....	465
server-member communication.....	453	inserting rule.....	477
unicast rekey configuration.....	454	log suppression.....	571
VPN group configuration.....	452	logging, overview.....	571
GTP		maximize-idp-sessions.....	569
access point name (APN) filtering.....	1024	packet capture.....	577
IMSI prefix filtering.....	1025	performance and capacity tuning.....	569
inspection objects.....	1016	policy.....	463
IP fragmentation.....	1023	policy, manage.....	464
policy-based.....	1016	policy, overview.....	463
GTP messages.....	1022	protocol decoder.....	512
length, filtering by.....	1019	rulebase, application-level DDoS.....	479
rate, limiting by.....	1022	rulebase, DDoS.....	479
type, filtering by.....	1019	rulebase, exempt.....	484
types.....	1019	rulebase, IPS.....	480
versions 0 and 1.....	1022	rulebase, overview.....	476
		rules, actions.....	473
H		rules, IP actions.....	475
hardware		rules, match conditions.....	471
supported platforms.....xl		rules, objects.....	471
hardware setup, chassis cluster.....	857, 863	rules, overview.....	470
hash-based message authentication code.....	359	send attack logs to the IC.....	576
heartbeats.....	834	setting terminal rules.....	488
group VPN.....	457	signature database.....	535
high availability.....	1015	terminal rules, overview.....	487
HMAC.....	359	verify load status.....	546
Host Checker See UAC, Host Checker policy		verify policy compilation.....	546
enforcement		verify signature database version.....	547
hub-and-spoke.....	375	IDP application-level DDoS	
		configuring statistic reporting.....	533
I		statistics reporting overview.....	531
ICMP		IDP policy	
floods.....	758, 759	application identification.....	551
fragments.....	736	overview.....	463
IPv6.....	53	rulebase, exempt.....	484
large packets.....	737	IDP, inline tap mode	
Path MTU.....	55	configuring.....	469
ICMP header flags.....	507	overview.....	468
IDP		IKE.....	357
application and services.....	494	gateway and peer authentication.....	370
application identification.....	549	Phase 1 proposals	
application sets.....	493	group VPN.....	450
application sets, configuring.....	495	predefined.....	367
custom attacks, properties.....	503, 508, 509	Phase 2 proposals	
deactivating rules.....	478	configuring.....	373
defining exempt rulebase.....	485	predefined.....	371
defining IPS rulebase.....	481	proxy IDs.....	371

-
- IMSI prefix filtering.....1025
 - in-service upgrade
 - chassis cluster.....874
 - Infranet agent *See* UAC, Infranet agent
 - Infranet Controller *See* UAC, Infranet Controller
 - Infranet Enforcer *See* UAC, Junos OS Enforcer
 - initiating manual redundancy group failover.....814
 - inline tap mode
 - overview.....468
 - inspections.....709
 - interface monitoring configuration.....805
 - interfaces.....86
 - control.....856
 - fabric.....856
 - interfaces on J Series devices
 - management.....855
 - node.....853
 - interfaces on SRX Series devices
 - management.....855
 - node.....846
 - intrusion detection and prevention *See* IDP
 - IP options
 - incorrectly formatted.....738
 - loose source route.....715
 - record route.....715, 717
 - security.....715, 717
 - source route.....730
 - stream ID.....715, 717
 - strict source route.....715
 - timestamp.....715, 717
 - IP packet fragments.....740
 - IP protocol header.....505
 - IP spoofing.....729
 - IPS rulebase
 - configuring.....481
 - IPsec
 - digital signature.....386
 - overview.....355
 - SAs.....355, 360, 371, 423
 - group VPN configuration.....451
 - See also* group VPNs
 - security protocols
 - Authentication Header (AH).....358
 - Encapsulating Security Protocol (ESP).....358
 - tunnel.....355
 - creating through dynamic VPN
 - feature.....405
 - tunnel mode.....361
 - tunnel negotiation.....360
 - UAC support.....332
 - IPv6
 - address examples.....48
 - address format.....48
 - address space.....46
 - address types.....46, 47
 - addressing.....46
 - anycast addresses.....47
 - basic packet header fields.....50
 - enabling.....61
 - features.....46
 - flow module sanity checks.....53
 - host-inbound traffic.....47
 - ICMP overview.....53
 - multicast addresses.....47
 - overview.....46
 - packet fragmentation.....57
 - packet header extension fields.....52
 - packet header overview.....49
 - Path MTU.....55
 - sessions.....57
 - SRX Series high-end devices.....57
 - unicast addresses.....47
 - J**
 - JUEP *See* UAC, device authentication
 - Junos OS Enforcer *See* UAC, Junos OS Enforcer
 - K**
 - KEK *See* group VPNs
 - key activation
 - group VPN.....431
 - L**
 - L2TP.....1015
 - land attack detection
 - configuration.....761
 - overview.....760
 - local certificate.....388
 - log suppression.....571
 - configuring.....574
 - logging
 - IDP, overview.....571
 - logging, traffic.....1016
 - loose source route IP detection
 - configuration.....715

M

main mode.....	368
management interfaces.....	855
configuring.....	867
manual key management	
overview.....	357
manuals	
comments on.....	xlii
MD5.....	359
Message Digest version 5 (MD5).....	359
MGCP ALG.....	261
commands.....	264
entities.....	262
security.....	262
MGCP timeouts	
inactivity.....	267
Mobile Station (MS) mode.....	1024
modes	
aggressive.....	368
main.....	368
tunnel.....	361
modes, operational	
NAT.....	1015
route.....	1015
transparent.....	1015
modes, selection	
APN.....	1024
Mobile Station (MS).....	1024
network.....	1024
verified.....	1024
modulus.....	358
MS RPC ALG, defined.....	291
multimedia sessions, SIP.....	207

N

NAT.....	927
address shifting.....	963
destination.....	941
destination NAT address pools.....	942
destination NAT configuration.....	944
destination NAT configuration examples.....	944
destination NAT overview.....	942
destination NAT rules.....	943
disabling port randomization.....	1002
overview.....	927
persistent addresses.....	964
persistent NAT.....	1003
persistent NAT configuration overview.....	1005
persistent NAT overview.....	1003

port address translation.....	962
proxy ARP.....	1008
rule sets and rules.....	928
source.....	959
source NAT address pools.....	961
source NAT configuration.....	965
source NAT configuration examples.....	965
source NAT overview.....	960
source NAT rules.....	964
static.....	930
static NAT configuration.....	932
static NAT configuration examples.....	932
static NAT overview.....	930
static NAT rules.....	931
STUN protocol.....	1004
verify configuration.....	1009
without port address translation.....	963
NAT mode.....	1015
Network Address Translation See NAT	
network mode.....	1024
node interfaces on J Series devices.....	853
node interfaces on SRX Series devices.....	846
notice icons.....	xl

O

Odyssey Access Client See UAC, Infranet agent	
operational modes	
NAT.....	1015
route.....	1015
transparent.....	1015

P

packet capture	
IDP.....	577
packet filtering.....	3, 69
packet fragmentation	
IPv6.....	57
packet processing.....	3, 69
stateful.....	3, 69
stateless.....	3, 69
packet-based processing.....	5, 71
parentheses, in syntax descriptions.....	xli
pass-through authentication.....	298
Path MTU	
Path MTU.....	55
Perfect Forward Secrecy See PFS	
PFS.....	372

-
- Phase 1.....367
 - proposals.....367
 - proposals, predefined.....367
 - Phase 2.....371
 - proposals.....371
 - proposals, configuring.....373
 - proposals, predefined.....371
 - ping of death attack protection
 - configuration.....763
 - overview.....762
 - pinholes.....210
 - PKI.....383
 - using SCEP.....388
 - policies.....1016
 - application services processing order.....332
 - core section.....726
 - schedulers
 - associating.....137
 - shadowing.....130
 - policies, configuring.....1016
 - policy
 - IDP See IDP
 - policy templates
 - predefined.....537
 - port scan attack protection
 - overview.....713
 - predefined attack objects.....540
 - predefined policy templates.....537
 - overview.....537
 - preshaed key.....357
 - probes
 - network.....711
 - open ports.....713
 - operating systems.....720, 724
 - processing
 - data.....3, 69
 - flow-based.....4, 70
 - packet-based.....5, 71
 - proposals
 - Phase 1.....367
 - Phase 2.....371
 - protocol anomaly.....508
 - protocol anomaly attack.....509
 - direction.....509
 - expression (boolean expression).....510
 - member index.....511
 - member index sample.....511
 - order.....510
 - reset.....510
 - sample.....509, 511
 - scope.....510
 - test condition.....509
 - protocol anomaly attack sample.....509
 - protocol binding.....500
 - sample format.....502
 - proxy IDs.....371
 - public/private key pair.....385
- R**
- rate limiting, GTP-C messages.....1022
 - reconnaissance
 - address sweep.....711
 - FIN scans.....725
 - IP options.....715
 - port scan.....713
 - SYN and FIN flags set.....720
 - TCP packet without flags.....724
 - reconnaissance deterrence
 - IP address sweeps.....711
 - blocking.....711
 - overview.....711
 - record route IP option.....715, 717
 - redundancy group
 - configuring dampening time between
 - back-to-back failovers.....816
 - initiating manual failover.....814
 - redundancy group configuration.....803
 - redundancy groups
 - about.....797
 - group 0.....798
 - groups 1 through 128.....799
 - interface monitoring.....804
 - IP address monitoring.....806
 - redundant Ethernet interface LAG.....822
 - configuration.....823
 - redundant Ethernet interfaces
 - configuring.....819
 - understanding.....817
 - registry changes, Access Manager.....417
 - rekey messages.....429
 - intervals.....429
 - types.....429
 - See also group VPNs
 - Remote Access Management Solution See dynamic VPNs

remote access server	
logging into for the first time.....	411
logging into subsequent sessions.....	412
overview.....	405
replay protection.....	372
reregistration	
group member.....	430
resource access policies <i>See</i> UAC, resource access	
policies	
reth	
link aggregation group.....	822
link aggregation group configuration.....	823
RFCs	
1038, Revised IP Security Option.....	715
791, Internet Protocol.....	715
793, Transmission Control Protocol.....	722
roles <i>See</i> UAC, user roles	
route mode.....	1015
RPC	
Sun RPC.....	288
rulebase	
exempt, attack objects.....	484
exempt, match condition.....	484
exempt, overview.....	484
IPS, action.....	480
IPS, attack objects.....	480
IPS, IP action.....	480
IPS, match condition.....	480
IPS, notification.....	480
IPS, overview.....	480
IPS, terminal flag.....	480
overview.....	476
rules.....	470
rules	
actions.....	473
deactivating.....	478
inserting.....	477
IP actions.....	475
match conditions.....	471
objects.....	471
objects, address.....	471
objects, attack.....	472
objects, service.....	471
objects, zone.....	471
overview.....	470
terminal.....	487
S	
SA parameters.....	360
SAs.....	371, 423
<i>See also</i> group VPNs	
SCCP	
allowing unknown message types.....	252, 253
configuring DoS attack protection.....	254, 255
setting inactive media timeout.....	251
SCEP.....	388, 391, 392
digital certificates.....	388
enrolling a local certificate.....	392
PKCS #10, PKCS #7.....	393
reenrolling certificates.....	396
RSA key.....	390
scope policies <i>See</i> group VPNs	
screen	
address sweep.....	711
bad IP options, drop.....	738
FIN with no ACK.....	726
FIN without ACK flag, drop.....	722
ICMP	
fragments, block.....	736
ICMP floods.....	758, 759
IP options.....	715
IP packet fragments, block.....	740
IP spoofing.....	729
Land attacks.....	760, 761
large ICMP packets, block.....	737
loose source route IP option, detect.....	730
Ping of Death.....	762
port scan.....	713
source route IP option, deny.....	730
strict source route IP option, detect.....	730
SYN and FIN flags set.....	720
SYN floods.....	748, 753
SYN fragments, detect.....	741
SYN-ACK-ACK proxy floods.....	747
TCP packet without flags, detect.....	724
Teardrop.....	763, 764
UDP floods.....	759, 760
unknown protocols, drop.....	739
WinNuke attacks.....	764, 765
Secure Hash Algorithm-1.....	359
SecurID.....	313
security checks, disabling TCP packet.....	76
security IP option.....	715, 717
security policy	
enabling IDP.....	465
security zones.....	85
creating.....	87
functional.....	87

host inbound traffic.....	90	verify load status.....	546
protocols.....	95	verify policy compilation.....	546
interfaces.....	86	verify version.....	547
ports.....	86	version, overview.....	541
TCP-reset parameter.....	98	<i>See also</i> IDP	
selection modes		SIP	
APN.....	1024	connection information.....	209
Mobile Station (MS).....	1024	defined.....	207
Network.....	1024	media announcements.....	209
verified.....	1024	messages.....	207
self-signed certificates		multimedia sessions.....	207
about.....	398	pinholes.....	208
automatically generated.....	399	request methods.....	212
manually generated.....	399, 400	response codes.....	228
sequence-number validation.....	1023	RTCP.....	209
service binding.....	497, 550	RTP.....	209
services		signaling.....	208
IDP, configuring.....	494	SIP ALG.....	210
timeout threshold.....	146	call duration and timeouts.....	213
session		SIP NAT	
changing characteristics.....	7, 73	call setup.....	221
controlling termination.....	74	defined.....	221
session limits.....	744	SIP timeouts	
source-based.....	744, 745	inactivity.....	213
session lookup.....	80	media inactivity.....	214, 251, 575
session table floods.....	726, 744	session inactivity.....	213
session-based processing.....	80	signaling inactivity.....	213
setting the node and cluster IDs.....	866	SNMP failover traps.....	816
SHA-1.....	359	source IP route attack protection	
signature attack sample.....	508	overview.....	730
signature custom attack.....	503	source NAT.....	959
context.....	503	address pools.....	961
direction.....	504	address shifting.....	963
ICMP header.....	507	address shifting configuration example.....	984
IP protocol flags.....	505	addresses with PAT configuration	
pattern.....	505	example.....	974
protocol-specific parameters.....	505	addresses without PAT configuration	
sample.....	508	example.....	979
TCP header.....	506	configuration overview.....	965
UDP header.....	507	disabling port randomization.....	1002
signature database.....	535	egress interface translation configuration	
attack object groups.....	540	example.....	966
automatic update.....	545	multiple rules configuration example.....	989
manually update.....	543	overview.....	960
overview.....	535	persistent addresses.....	964
predefined attack objects.....	540	persistent NAT.....	1003
predefined policy templates.....	537	persistent NAT configuration overview.....	1005
updating, overview.....	542	persistent NAT overview.....	1003
verify.....	545	port address translation.....	962

rules.....	964	SYN fragment protection	
single address translation configuration		overview.....	741
example.....	969	SYN-ACK-ACK proxy floods.....	747
STUN protocol.....	1004	SYN-ACK-ACK-proxy flood protection	
with destination NAT configuration		configuration.....	747
example.....	996	syntax conventions.....	xl
without port address translation.....	963		
SPUs		T	
monitoring.....	810	TCP header flag attack protection	
stateful.....	709	configuration.....	506
stateful and stateless data processing.....	3, 69	overview.....	724
stateful inspection.....	709	teardrop attack protection	
stateful packet processing	3, 69	configuration.....	764
stateless firewall filters.....	6, 72	overview.....	763
stateless packet processing.....	3, 69	technical support	
static NAT.....	930	contacting JTAC.....	xlii
configuration overview.....	932	TEK See group VPNs	
overview.....	930	terminal rules	
rules.....	931	overview.....	487
single address translation configuration		setting.....	488
example.....	932	terminology	
subnet translation configuration		IDP policy.....	464
example.....	936	three-way handshakes.....	748
statistics		time binding.....	502
application-level DDoS configuring.....	533	count.....	503
application-level DDoS overview.....	531	scope.....	502
statistics, verifying		timestamp IP option.....	715, 717
for application identification.....	556, 784	traffic	
stream ID IP option.....	715, 717	counting.....	1016
strict source route IP option.....	715	logging.....	1016
Sun RPC ALG.....	288	transparent mode.....	1015
call scenarios.....	288	transport mode.....	361
defined.....	288	Triple DES.....	359
support, technical See technical support		tunnel mode	
SYN and FIN flags protection		overview.....	361
overview.....	720		
SYN checking.....	726	U	
asymmetric routing.....	726	UAC	
reconnaissance hole.....	726	authentication tables	
session table floods.....	726	failover processing.....	351
SYN cookies.....	756	overview.....	332
SYN floods.....	748, 753	captive portal.....	344
alarm threshold.....	751	See also captive portal	
attack threshold.....	751	certificates See UAC, device authentication	
destination threshold.....	751	clustering See UAC, failover processing	
source threshold.....	751	device authentication	
SYN cookies.....	756	configuring.....	330
threshold.....	749	overview.....	329
timeout.....	751	dynamic auth table provisioning.....	332

- failover processing
 - configuring timeout actions.....351
 - connecting to cluster.....330
 - overview.....351
 - Host Checker policy enforcement.....342
 - Infranet agent
 - agentless access.....342
 - Odyssey Access Client.....342
 - overview.....327, 342
 - support information.....342
 - Infranet Controller
 - communications with Junos OS
 - Enforcer.....329
 - configuring access to.....330
 - overview.....327
 - IPsec support.....332
 - JUEP *See* UAC, device authentication
 - Junos OS Enforcer
 - communications with Infranet
 - Controller.....329
 - enabling.....330
 - overview.....327
 - logging.....333
 - overview.....327
 - policies
 - application services processing
 - order.....332
 - enforcement overview.....332
 - resource access policies
 - failover processing.....351
 - overview.....332
 - show commands.....332
 - test-only mode.....333
 - timeout actions *See* UAC, failover processing
 - user roles.....332
- UDP header attack protection
 - configuration.....507
 - Unified Access Control *See* UAC
 - Unified Threat Management
 - antispam filtering.....587
 - antivirus protection, express.....649
 - antivirus protection, full.....605
 - content filtering.....665
 - licensing.....583
 - overview.....581
 - platform support.....583
 - web filtering.....679
 - unknown protocol attack protection
 - overview.....739
 - upgrading
 - chassis cluster ISSU.....874
 - user roles *See* UAC, user roles
 - UTM
 - WELF support for log files.....583
- ## V
- verification
 - application system cache.....556, 559, 784
 - verified mode.....1024
 - verifying
 - chassis cluster interfaces.....821
 - chassis cluster redundancy group status.....804
 - chassis cluster statistics.....870
 - chassis cluster status.....872
 - chassis clusters.....869
 - IDP policy compilation.....546
 - IDP policy load status.....546
 - signature database.....545
 - signature database version.....547
 - version
 - signature database.....541
 - VPNs
 - aggressive mode.....368
 - AutoKey IKE.....357
 - Diffie-Hellman exchange.....358
 - Diffie-Hellman groups.....358
 - dynamic VPN *See* dynamic VPNs
 - global settings.....375
 - group *See* group VPNs
 - group configuration.....452
 - group VPN *See* group VPNs
 - main mode.....368
 - Phase 1.....367
 - Phase 2.....371
 - replay protection.....372
- ## W
- Web Filtering.....679
 - cache.....681
 - integrated.....680
 - local.....700
 - profiles.....681, 701
 - verifying.....704
 - wildcards.....1024
 - Windows registry changes, Access Manager.....417
 - WinNuke attack protection
 - configuration.....765
 - overview.....764

Z

zombie agents.....	743
zones	
functional.....	87
security.....	85