

Stateless Firewall Filter Configuration



Published: 2011-06-29
Revision

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.

Stateless Firewall Filter Configuration
Copyright © 2011, Juniper Networks, Inc.
All rights reserved.

Revision History
May 2011—Revision 1; initial release

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

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)).

Table of Contents

Part 1

Overview

Chapter 1

Introduction to Stateless Firewall Filters 3

Router Data Flow Overview 3

Flow of Routing Information 3

Flow of Data Packets 3

Flow of Local Packets 4

Interdependent Flows of Routing Information and Packets 4

Stateless Firewall Filter Overview 5

Packet Flow Control 5

Data Packet Flow Control 5

Local Packet Flow Control 5

Stateless and Stateful Firewall Filters 5

Purpose of Stateless Firewall Filters 5

Stateless Firewall Filter Types 6

Standard Stateless Firewall Filters 6

Service Filters 6

Simple Filters 7

Stateless Firewall Filter Components 7

Protocol Family 7

Filter Type 8

Terms 9

Match Conditions 9

Actions 10

Filter-Terminating Actions 10

Nonterminating Actions 10

Flow Control Action 10

Stateless Firewall Filter Application Points 11

Chapter 2

Standard Firewall Filter Overview 15

Standard Stateless Firewall Filter Overview 15

How Standard Firewall Filters Evaluate Packets 16

Firewall Filters That Contain a Single Term 16

Firewall Filters That Contain Multiple Terms 16

Firewall Filter Terms That Do Not Contain Any Match Conditions 17

Firewall Filter Terms That Do Not Contain Any Actions 17

Firewall Filter Default Action 17

Guidelines for Configuring Standard Firewall Filters 17

Statement Hierarchy for Configuring Standard Firewall Filters 17

Standard Firewall Filter Protocol Families 18

Standard Firewall Filter Names and Options 19

	Standard Firewall Filter Terms	19
	Standard Firewall Filter Match Conditions	19
	Standard Firewall Filter Actions	21
	Guidelines for Applying Standard Firewall Filters	22
	Applying Standard Firewall Filters Overview	22
	Applying a Firewall Filter to a Router's Physical Interfaces	22
	Applying a Firewall Filter to the Router's Loopback Interface	22
	Applying a Firewall Filter to Multiple Interfaces	22
	Statement Hierarchy for Applying Standard Firewall Filters	22
	Restrictions on Applying Standard Firewall Filters	23
	Number of Input and Output Filters Per Logical Interface	23
	MPLS and Layer 2 CCC Firewall Filters in Lists	23
	Layer 2 CCC Firewall Filters on MX Series Routers	24
	Protocol-Independent Firewall Filters on the Loopback Interface	24
	Basic Uses for Standard Firewall Filters	24
	Using Standard Firewall Filters to Affect Local Packets	24
	Trusted Sources	24
	Flood Prevention	25
	Using Standard Firewall Filters to Affect Data Packets	25
Chapter 3	Standard Firewall Filter Match Conditions Overview	27
	Firewall Filter Match Conditions Based on Numbers or Text Aliases	27
	Matching on a Single Numeric Value	27
	Matching on a Range of Numeric Values	27
	Matching on a Text Alias for a Numeric Value	28
	Matching on a List of Numeric Values or Text Aliases	28
	Firewall Filter Match Conditions Based on Bit-Field Values	28
	Match Conditions for Bit-Field Values	28
	Match Conditions for Common Bit-Field Values or Combinations	29
	Logical Operators for Bit-Field Values	30
	Matching on a Single Bit-Field Value or Text Alias	30
	Matching on Multiple Bit-Field Values or Text Aliases	31
	Matching on a Negated Bit-Field Value	31
	Matching on the Logical OR of Two Bit-Field Values	31
	Matching on the Logical AND of Two Bit-Field Values	32
	Grouping Bit-Field Match Conditions	32
	Firewall Filter Match Conditions Based on Address Fields	32
	Implied Match on the '0/0' Address for Firewall Filter Match Conditions	
	Based on Address Fields	33
	Matching an Address Field to a Subnet Mask or Prefix	33
	IPv4 Subnet Mask Notation	33
	Prefix Notation	33
	Default Prefix Length for IPv4 Addresses	33
	Default Prefix Length for IPv6 Addresses	33
	Default Prefix Length for MAC Addresses	33
	Matching an Address Field to an Excluded Value	34
	Excluding IP Addresses in IPv4 or IPv6 Traffic	34
	Excluding IP Addresses in VPLS or Layer 2 Bridging Traffic	35
	Excluding MAC Addresses in VPLS or Layer 2 Bridging Traffic	35

	Matching Either IP Address Field to a Single Value	35
	Matching Either IP Address Field in IPv4 or IPv6 Traffic	36
	Matching Either IP Address Field in VPLS or Layer 2 Bridging Traffic	36
	Matching an Address Field to Noncontiguous Prefixes	36
	Matching an Address Field to a Prefix List	38
	Firewall Filter Match Conditions Based on Address Classes	39
	Source-Class Usage	39
	Destination-Class Usage	39
	Guidelines for Applying SCU or DCU Firewall Filters to Output Interfaces	39
Chapter 4	Introduction to Standard Firewall Filters for Fragment Handling	41
	Firewall Filters That Handle Fragmented Packets Overview	41
Chapter 5	Introduction to Standard Firewall Configuration	43
	Stateless Firewall Filters That Reference Policers Overview	43
	Multiple Standard Firewall Filters Applied as a List Overview	44
	The Challenge: Simplify Large-Scale Firewall Filter Administration	44
	A Solution: Apply Lists of Firewall Filters	45
	Configuration of Multiple Filters for Filter Lists	45
	Application of Filter Lists to a Router Interface	45
	Interface-Specific Names for Filter Lists	46
	How Filter Lists Evaluate Packets	46
	Guidelines for Applying Multiple Standard Firewall Filters as a List	47
	Statement Hierarchy for Applying Lists of Multiple Firewall Filters	47
	Filter Input Lists and Output Lists for Router Interfaces	47
	Types of Filters Supported in Lists	47
	Restrictions on Applying Filter Lists for MPLS or Layer 2 CCC Traffic	48
	Multiple Standard Firewall Filters in a Nested Configuration Overview	48
	The Challenge: Simplify Large-Scale Firewall Filter Administration	48
	A Solution: Configure Nested References to Firewall Filters	48
	Configuration of Nested Firewall Filters	49
	Application of Nested Firewall Filters to a Router Interface	49
	Guidelines for Nesting References to Multiple Standard Firewall Filters	49
	Statement Hierarchy for Configuring Nested Firewall Filters	49
	Filter-Defining Terms and Filter-Referencing Terms	50
	Types of Filters Supported in Nested Configurations	50
	Number of Filter References in a Single Filter	50
	Depth of Filter Nesting	50
	Interface-Specific Firewall Filter Instances Overview	51
	Instantiation of Interface-Specific Firewall Filters	51
	Interface-Specific Names for Firewall Filter Instances	51
	Interface-Specific Firewall Filter Counters	52
	Interface-Specific Firewall Filter Policers	52
	Filtering Packets Received on a Set of Interface Groups Overview	52
	Filtering Packets Received on an Interface Set Overview	53
	Filter-Based Forwarding Overview	53
	Filters That Classify Packets or Direct Them to Routing Instances	54
	Input Filtering to Classify and Forward Packets Within the Router	54
	Output Filtering to Forward Packets to Another Routing Table	54
	Restrictions for Applying Filter-Based Forwarding	54

	Accounting for Standard Firewall Filters Overview	55
	System Logging Overview	55
	System Logging of Events Generated for the Firewall Facility	56
	Logging of Packet Headers Evaluated by a Firewall Filter Term	58
Chapter 6	Introduction to Service Filter Configuration	61
	Service Filter Overview	61
	Services	61
	Service Rules	61
	Service Rule Refinement	61
	Service Filter Counters	62
	How Service Filters Evaluate Packets	62
	Service Filters That Contain a Single Term	63
	Service Filters That Contain Multiple Terms	63
	Service Filter Terms That Do Not Contain Any Match Conditions	63
	Service Filter Terms That Do Not Contain Any Actions	63
	Service Filter Default Action	63
	Guidelines for Configuring Service Filters	64
	Statement Hierarchy for Configuring Service Filters	64
	Service Filter Protocol Families	64
	Service Filter Names	64
	Service Filter Terms	64
	Service Filter Match Conditions	65
	Service Filter Terminating Actions	65
	Guidelines for Applying Service Filters	66
	Restrictions for Adaptive Services Interfaces	66
	Adaptive Services Interfaces	66
	System Logging to a Remote Host from M Series Routers	66
	Statement Hierarchy for Applying Service Filters	66
	Associating Service Rules with Adaptive Services Interfaces	67
	Filtering Traffic Before Accepting Packets for Service Processing	67
	Postservice Filtering of Returning Service Traffic	68
Chapter 7	Introduction to Simple Filter Configuration	69
	Simple Filter Overview	69
	How Simple Filters Evaluate Packets	69
	Simple Filters That Contain a Single Term	69
	Simple Filters That Contain Multiple Terms	70
	Simple Filter Terms That Do Not Contain Any Match Conditions	70
	Simple Filter Terms That Do Not Contain Any Actions	70
	Simple Filter Default Action	70
	Guidelines for Configuring Simple Filters	70
	Statement Hierarchy for Configuring Simple Filters	71
	Simple Filter Protocol Families	71
	Simple Filter Names	71
	Simple Filter Terms	71
	Simple Filter Match Conditions	72
	Simple Filter Terminating Actions	73
	Simple Filter Nonterminating Actions	73

	Guidelines for Applying Simple Filters	74
	Statement Hierarchy for Applying Simple Filters	74
	Restrictions for Applying Simple Filters	74
Chapter 8	Introduction to Firewall Filter Configuration in Logical Systems	77
	Stateless Firewall Filters in Logical Systems Overview	77
	Logical Systems	77
	Stateless Firewall Filters in Logical Systems	77
	Identifiers for Firewall Objects in Logical Systems	77
	Guidelines for Configuring and Applying Firewall Filters in Logical Systems	78
	Statement Hierarchy for Configuring Firewall Filters in Logical Systems	78
	Filter Types in Logical Systems	79
	Firewall Filter Protocol Families in Logical Systems	79
	Firewall Filter Match Conditions in Logical Systems	79
	Firewall Filter Actions in Logical Systems	80
	Statement Hierarchy for Applying Firewall Filters in Logical Systems	80
	References from a Firewall Filter in a Logical System to Subordinate Objects . . .	81
	Resolution of References from a Firewall Filter to Subordinate Objects	81
	Valid Reference from a Firewall Filter to a Subordinate Object	81
	References from a Firewall Filter in a Logical System to Nonfirewall Objects	82
	Resolution of References from a Firewall Filter to Nonfirewall Objects	82
	Valid Reference to a Nonfirewall Object Outside of the Logical System	82
	References from a Nonfirewall Object in a Logical System to a Firewall Filter . . .	84
	Resolution of References from a Nonfirewall Object to a Firewall Filter	84
	Invalid Reference to a Firewall Filter Outside of the Logical System	85
	Valid Reference to a Firewall Filter Within the Logical System	86
	Valid Reference to a Firewall Filter Outside of the Logical System	88
Part 2	Configuration	
Chapter 9	Standard Firewall Filter Configurations That Match Packets	93
	Example: Configuring a Filter to Match on IPv6 Flags	93
	Example: Configuring a Filter to Match on Port and Protocol Fields	94
	Example: Configuring a Filter to Match on Two Unrelated Criteria	97
	Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List	100
Chapter 10	Standard Firewall Filters That Count Packets	105
	Example: Configuring a Filter to Count Accepted and Rejected Packets	105
	Example: Configuring a Filter to Count and Discard IP Options Packets	108
	Example: Configuring a Filter to Count IP Options Packets	111
Chapter 11	Standard Firewall Filters That Act on Packets	117
	Example: Configuring a Filter to Set the DSCP Bit to Zero	117
	Example: Configuring a Filter to Count and Sample Accepted Packets	120
Chapter 12	Standard Firewall Filters for Trusted Sources	125
	Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources	125
	Example: Configuring a Filter to Block Telnet and SSH Access	130
	Example: Configuring a Filter to Block TFTP Access	136

	Example: Configuring a Filter to Accept OSPF Packets from a Prefix	139
	Example: Configuring a Filter to Accept DHCP Packets Based on Address	141
	Example: Configuring a Filter to Block TCP Access to a Port Except From Specified BGP Peers	144
Chapter 13	Standard Firewall Filters for Flood Prevention	151
	Example: Configuring a Stateless Firewall Filter to Protect Against TCP and ICMP Floods	151
	Example: Configuring a Filter to Accept Packets Based on IPv6 TCP Flags	158
Chapter 14	Standard Firewall Filters for Fragment Handling	161
	Example: Configuring a Stateless Firewall Filter to Handle Fragments	161
Chapter 15	Standard Firewall Filters for Setting Rate Limits	167
	Example: Configuring a Rate-Limiting Filter Based on Destination Class	167
Chapter 16	Standard Firewall Configuration	171
	Example: Applying Lists of Multiple Standard Firewall Filters	171
	Example: Nesting References to Multiple Standard Firewall Filters	176
	Example: Configuring Interface-Specific Firewall Filter Counters	180
	Example: Filtering Packets Received on an Interface Group	184
	Example: Filtering Packets Received on an Interface Set	188
	Example: Configuring Filter-Based Forwarding on the Source Address	194
Chapter 17	Standard Firewall Configuration Options	201
	Example: Configuring Statistics Collection for a Standard Firewall Filter	201
	Example: Configuring Logging for a Stateless Firewall Filter Term	206
Chapter 18	Service Filter Configuration	211
	Example: Configuring and Applying Service Filters	211
Chapter 19	Simple Filter Configuration	217
	Example: Configuring and Applying a Simple Filter	217
Chapter 20	Firewall Filter Configuration in Logical Systems	223
	Example: Configuring a Stateless Firewall Filter to Protect a Logical System Against ICMP Floods	223
Part 3	Administration	
Chapter 21	Firewall Filter Standards	229
	Supported Standards for Filtering	229
Chapter 22	Firewall Filter Reference	231
	Using the CLI Editor in Configuration Mode	231
Chapter 23	Standard Firewall Filter Match Conditions and Actions	235
	Standard Firewall Filter Match Conditions for Protocol-Independent Traffic	235
	Standard Firewall Filter Match Conditions for IPv4 Traffic	236
	Standard Firewall Filter Match Conditions for IPv6 Traffic	245
	Standard Firewall Filter Match Conditions for MPLS Traffic	250

	Standard Firewall Filter Match Conditions for MPLS-Tagged IPv4 Traffic	252
	Matching on IPv4 Packet Header Address or Port Fields in MPLS Flows	252
	IP Address Match Conditions for MPLS Traffic	253
	IP Port Match Conditions for MPLS Traffic	253
	Standard Firewall Filter Match Conditions for VPLS Traffic	254
	Standard Firewall Filter Match Conditions for Layer 2 CCC Traffic	261
	Standard Firewall Filter Match Conditions for Layer 2 Bridging Traffic	263
	Standard Firewall Filter Terminating Actions	269
	Standard Firewall Filter Nonterminating Actions	270
Chapter 24	Service Filter Match Conditions and Actions	275
	Service Filter Match Conditions for IPv4 or IPv6 Traffic	275
	Service Filter Terminating Actions	281
	Service Filter Nonterminating Actions	282
Chapter 25	Reference Information for Firewall Filters in Logical Systems	283
	Unsupported Firewall Filter Statements for Logical Systems	283
	Unsupported Actions for Firewall Filters in Logical Systems	285
Chapter 26	Firewall Filter Statement Hierarchies	289
	Statement Hierarchy for Configuring Interface-Specific Firewall Filters	289
	Statement Hierarchy for Applying Interface-Specific Firewall Filters	290
	Statement Hierarchy for Assigning Interfaces to Interface Groups	291
	Statement Hierarchy for Configuring a Filter to Match on a Set of Interface Groups	291
	Statement Hierarchy for Applying Filters to an Interface Group	292
	Statement Hierarchy for Defining an Interface Set	293
	Statement Hierarchy for Configuring a Filter to Match on an Interface Set	293
	Statement Hierarchy for Configuring FBF for IPv4 or IPv6 Traffic	294
	Statement Hierarchy for Configuring FBF for MPLS-Tagged IPv4 Traffic	295
	Matching on IPv4 Address Fields	295
	Matching on TCP Port Number Fields	295
	Matching on UDP Port Number Fields	296
	Statement Hierarchy for Configuring Routing Instances for FBF	297
	Statement Hierarchy for Applying FBF Filters to Interfaces	298
	Statement Hierarchy for Configuring Firewall Filter Accounting Profiles	299
	Statement Hierarchy for Applying Firewall Filter Accounting Profiles	300
Chapter 27	Summary of Firewall Filter Configuration Statements	301
	accounting-profile	301
	family	302
	filter (Applying to a Logical Interface)	303
	filter (Configuring)	304
	firewall	305
	interface-set	305
	interface-specific	306
	prefix-list	307
	service-filter	308
	simple-filter	309
	term	310

Part 4

Index

Index	315
-------------	-----

List of Figures

Part 1	Overview	
Chapter 1	Introduction to Stateless Firewall Filters	3
	Figure 1: Flows of Routing Information and Packets	4
Part 2	Configuration	
Chapter 12	Standard Firewall Filters for Trusted Sources	125
	Figure 2: Typical Network with BGP Peer Sessions	145
Chapter 20	Firewall Filter Configuration in Logical Systems	223
	Figure 3: Logical System with a Stateless Firewall	224

List of Tables

Part 1	Overview	
Chapter 1	Introduction to Stateless Firewall Filters	3
	Table 1: Firewall Filter Protocol Families	7
	Table 2: Filter Types	8
	Table 3: Stateless Firewall Filter Configuration and Application Summary	11
Chapter 2	Standard Firewall Filter Overview	15
	Table 4: Standard Firewall Filter Match Conditions by Protocol Family	20
	Table 5: Standard Firewall Filter Action Categories	21
Chapter 3	Standard Firewall Filter Match Conditions Overview	27
	Table 6: Binary and Bit-Field Match Conditions for Firewall Filters	29
	Table 7: Bit-Field Match Conditions for Common Combinations	29
	Table 8: Bit-Field Logical Operators	30
Chapter 5	Introduction to Standard Firewall Configuration	43
	Table 9: Syslog Message Destinations for the Firewall Facility	57
	Table 10: Packet-Header Logs for Stateless Firewall Filter Terms	59
Chapter 7	Introduction to Simple Filter Configuration	69
	Table 11: Simple Filter Match Conditions	72
Part 3	Administration	
Chapter 23	Standard Firewall Filter Match Conditions and Actions	235
	Table 12: Standard Firewall Filter Match Conditions for Protocol-Independent Traffic	236
	Table 13: Standard Firewall Filter Match Conditions for IPv4 Traffic	236
	Table 14: Standard Firewall Filter Match Conditions for IPv6 Traffic	245
	Table 15: Standard Firewall Filter Match Conditions for MPLS Traffic	251
	Table 16: IP Address-Specific Firewall Filter Match Conditions for MPLS Traffic	253
	Table 17: IP Port-Specific Firewall Filter Match Conditions for MPLS Traffic	254
	Table 18: Standard Firewall Filter Match Conditions for VPLS Traffic	255
	Table 19: Standard Firewall Filter Match Conditions for Layer 2 CCC Traffic	262
	Table 20: Standard Firewall Filter Match Conditions for Layer 2 Bridging (MX Series 3D Universal Edge Routers Only)	263
	Table 21: Terminating Actions for Standard Firewall Filters	269
	Table 22: Nonterminating Actions for Standard Firewall Filters	271
Chapter 24	Service Filter Match Conditions and Actions	275
	Table 23: Service Filter Match Conditions for IPv4 or IPv6 Traffic	275

	Table 24: Terminating Actions for Service Filters	281
	Table 25: Nonterminating Actions for Service Filters	282
Chapter 25	Reference Information for Firewall Filters in Logical Systems	283
	Table 26: Unsupported Firewall Statements for Logical Systems	283
	Table 27: Unsupported Actions for Firewall Filters in Logical Systems	285

PART 1

Overview

- [Introduction to Stateless Firewall Filters on page 3](#)
- [Standard Firewall Filter Overview on page 15](#)
- [Standard Firewall Filter Match Conditions Overview on page 27](#)
- [Introduction to Standard Firewall Filters for Fragment Handling on page 41](#)
- [Introduction to Standard Firewall Configuration on page 43](#)
- [Introduction to Service Filter Configuration on page 61](#)
- [Introduction to Simple Filter Configuration on page 69](#)
- [Introduction to Firewall Filter Configuration in Logical Systems on page 77](#)

CHAPTER 1

Introduction to Stateless Firewall Filters

- Router Data Flow Overview on page 3
- Stateless Firewall Filter Overview on page 5
- Stateless Firewall Filter Types on page 6
- Stateless Firewall Filter Components on page 7
- Stateless Firewall Filter Application Points on page 11

Router Data Flow Overview

The Junos OS provides a *policy framework*, which is a collection of Junos OS policies that enable you to control flows of routing information and packets within the router.

- Flow of Routing Information on page 3
- Flow of Data Packets on page 3
- Flow of Local Packets on page 4
- Interdependent Flows of Routing Information and Packets on page 4

Flow of Routing Information

Routing information is the information about routes learned by the routing protocols from a router's neighbors. This information is stored in routing tables. The routing protocols advertise active routes only from the routing tables. An *active route* is a route that is chosen from all routes in the routing table to reach a destination.

To control which routes the routing protocols place in the routing tables and which routes the routing protocols advertise from the routing tables, you can configure *routing policies*, which are sets of rules that the policy framework uses to preempt default routing policies.

The Routing Engine, which is the router's control plane, handles the flow of routing information between the routing protocols and the routing tables and between the routing tables and the forwarding table. The Routing Engine runs the Junos OS and routing policies and stores the active router configuration, the master routing table, and the master forwarding table,

Flow of Data Packets

Data packets are chunks of data that transit the router as they are being forwarded from a source to a destination. When a router receives a data packet on an interface, it determines where to forward the packet by looking in the forwarding table for the best

route to a destination. The router then forwards the data packet toward its destination through the appropriate interface.

The Packet Forwarding Engine, which is the router's forwarding plane, handles the flow of data packets in and out of the router's physical interfaces. Although the Packet Forwarding Engine contains Layer 3 and Layer 4 header information, it does not contain the packet data itself (the packet's payload).

Flow of Local Packets

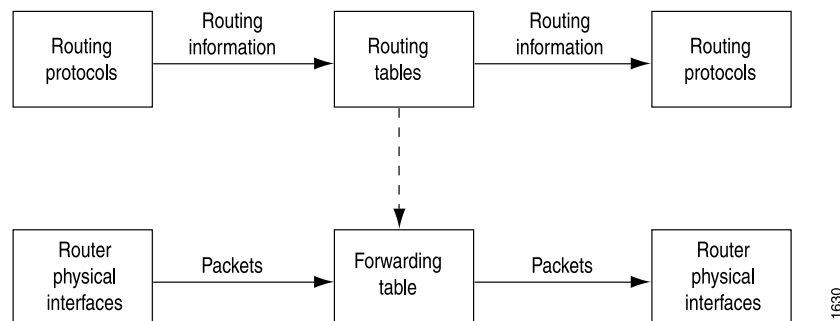
Local packets are chunks of data that are destined for or sent by the router. Local packets usually contain routing protocol data, data for IP services such as Telnet or SSH, and data for administrative protocols such as the Internet Control Message Protocol (ICMP). When the Routing Engine receives a local packet, it forwards the packet to the appropriate process or to the kernel, which are both part of the Routing Engine, or to the Packet Forwarding Engine.

The Routing Engine handles the flow of local packets from the router's physical interfaces and to the Routing Engine.

Interdependent Flows of Routing Information and Packets

Figure 1 on page 4 illustrates the flow of data through a router. Although routing information flows and packet flows are very different from one another, they are also interdependent.

Figure 1: Flows of Routing Information and Packets



Routing policies determine which routes the Routing Engine places in the forwarding table. The forwarding table, in turn, has an integral role in determining the appropriate physical interface through which to forward a packet.

Related Documentation

- Stateless Firewall Filter Overview on page 5
- "Packet Flow Within Routers Overview" in the [Junos OS Class of Service Configuration Guide](#)
- "How the Active Route Is Determined" in the [Junos OS Routing Protocols Configuration Guide](#)
- "Default Route Preference Values" in the [Junos OS Routing Protocols Configuration Guide](#)
- "Routing Policy Overview" in the [Junos OS Routing Policy Configuration Guide](#)

Stateless Firewall Filter Overview

This topic covers the following information:

- Packet Flow Control on page 5
- Stateless and Stateful Firewall Filters on page 5
- Purpose of Stateless Firewall Filters on page 5

Packet Flow Control

To influence which packets are allowed to transit the system and to apply special actions to packets as necessary, you can configure *stateless firewall filters*. A stateless firewall specifies a sequence of one or more packet-filtering rules, called *filter terms*. A filter term specifies *match conditions* to use to determine a match and *actions* to take on a matched packet. A stateless firewall filter enables you to manipulate any packet of a particular protocol family, including fragmented packets, based on evaluation of Layer 3 and Layer 4 header fields. You typically apply a stateless firewall filter to one or more interfaces that have been configured with protocol family features. You can apply a stateless firewall filter to an ingress interface, an egress interface, or both.

Data Packet Flow Control

To control the flow of data packets transiting the device as the packets are being forwarded from a source to a destination, you can apply stateless firewall filters to the input or output of the router's physical interfaces.

To enforce a specified bandwidth and maximum burst size for traffic sent or received on an interface, you can configure *policers*. Policers are a specialized type of stateless firewall filter and a primary component of the Junos OS *class-of-service* (CoS).

Local Packet Flow Control

To control the flow of local packets between the physical interfaces and the Routing Engine, you can apply stateless firewall filters to the input or output of the *loopback interface*. The loopback interface (**lo0**) is the interface to the Routing Engine and carries no data packets.

Stateless and Stateful Firewall Filters

A stateless firewall filter, also known as an *access control list* (ACL), does not statefully inspect traffic. Instead, it evaluates packet contents statically and does not keep track of the state of network connections. In contrast, a *stateful firewall filter* uses connection state information derived from other applications and past communications in the data flow to make dynamic control decisions.

The [Junos OS Firewall Filter and Policier Configuration Guide](#) describes *stateless firewall filters* supported on T Series, M Series, and MX Series routers. For information about Junos OS stateful firewall policies for J Series and SRX Series security devices, see "Security Policies Overview" in the [Junos OS Security Configuration Guide](#).

Purpose of Stateless Firewall Filters

The basic purpose of a stateless firewall filter is to enhance security through the use of packet filtering. Packet filtering enables you to inspect the components of incoming or

outgoing packets and then perform the actions you specify on packets that match the criteria you specify. The typical use of a stateless firewall filter is to protect the Routing Engine processes and resources from malicious or untrusted packets.

**Related
Documentation**

- Router Data Flow Overview on page 3
- Stateless Firewall Filter Types on page 6
- “Traffic Policing Overview” in the *Junos OS Firewall Filter and Policer Configuration Guide*
- “Packet Flow Through the CoS Process Overview” in the *Junos OS Class of Service Configuration Guide*

Stateless Firewall Filter Types

This topic covers the following information:

- Standard Stateless Firewall Filters on page 6
- Service Filters on page 6
- Simple Filters on page 7

Standard Stateless Firewall Filters

The Junos OS standard stateless firewall filters support a rich set of packet-matching criteria that you can use to match on specific traffic and perform specific actions, such as forwarding or dropping packets that match the criteria you specify. You can configure firewall filters to protect the local router or to protect another device that is either directly or indirectly connected to the local router. For example, you can use the filters to restrict the local packets that pass from the router’s physical interfaces to the Routing Engine. Such filters are useful in protecting the IP services that run on the Routing Engine, such as Telnet, SSH, and BGP, from denial-of-service attacks.



NOTE: If you configured targeted broadcast for virtual routing and forwarding (VRF) by including the `forward-and-send-to-re` statement, any firewall filter that is configured on the Routing Engine loopback interface (lo0) cannot be applied to the targeted broadcast packets that are forwarded to the Routing Engine. This is because broadcast packets are forwarded as flood next hop traffic and not as local next hop traffic, and you can only apply a firewall filter to local next hop routes for traffic directed toward the Routing Engine.

Service Filters

A service filter defines packet-filtering (a set of match conditions and a set of actions) for IPv4 or IPv6 traffic. You can apply a service filter to the inbound or outbound traffic at an adaptive services interface to perform packet filtering on traffic before it is accepted for service processing. You can also apply a service filter to the traffic that is returning to the services interface after service processing to perform postservice processing.

Service filters filter IPv4 and IPv6 traffic only and can be applied to logical interfaces on Adaptive Services PICs, MultiServices PICs, and MultiServices DPCs only. Service filters are not supported on J Series devices and Branch SRX devices.

Simple Filters

Simple filters are supported on Gigabit Ethernet intelligent queuing (IQ2) and Enhanced Queuing Dense Port Concentrator (EQ DPC) interfaces only. Unlike standard filters, simple filters support IPv4 traffic only and have a number of restrictions. For example, you cannot configure a terminating action for a simple filter. Simple filters always accept packets. Also, simple filters can be applied only as input filters. They are not supported on outbound traffic. Simple filters are recommended for metropolitan Ethernet applications.

Related Documentation

- Stateless Firewall Filter Overview on page 5
- Stateless Firewall Filter Components on page 7

Stateless Firewall Filter Components

This topic covers the following information:

- Protocol Family on page 7
- Filter Type on page 8
- Terms on page 9
- Match Conditions on page 9
- Actions on page 10

Protocol Family

Under the **firewall** statement, you can specify the protocol family for which you want to filter traffic.

Table 1 on page 7 describes the firewall filter protocol families.

Table 1: Firewall Filter Protocol Families

Type of Traffic to Be Filtered	Protocol Family Configuration Statement	Comments
Protocol Independent	family any	All protocol families configured on a logical interface.
Internet Protocol version 4 (IPv4)	family inet	The family inet statement is optional for IPv4.
Internet Protocol version 6 (IPv6)	family inet6	
MPLS	family mpls	
MPLS-tagged IPv4	family mpls	Supports matching on IP addresses and ports, up to five MPLS stacked labels.
Virtual private LAN service (VPLS)	family vpls	

Table 1: Firewall Filter Protocol Families (*continued*)

Type of Traffic to Be Filtered	Protocol Family Configuration Statement	Comments
Layer 2 Circuit Cross-Connection	family ccc	
Layer 2 Bridging	family bridge	MX Series routers only.

Filter Type

Under the **family *family-name*** statement, you can specify the type and name of the filter you want to configure.

Table 2 on page 8 describes the firewall filter types.

Table 2: Filter Types

Filter Type	Filter Configuration Statement	Description
Stateless Firewall Filter	filter <i>filter-name</i>	<p>Filters the following traffic types:</p> <ul style="list-style-type: none"> • Protocol independent • IPv4 • IPv6 • MPLS • MPLS-tagged IPv4 • VPLS • Layer 2 CCC • Layer 2 bridging (MX Series routers only)
Service Filter	service-filter <i>service-filter-name</i>	<p>Defines packet-filtering to be applied to ingress or egress before it is accepted for service processing or applied to returning service traffic after service processing has completed.</p> <p>Filters the following traffic types:</p> <ul style="list-style-type: none"> • IPv4 • IPv6 <p>Supported at logical interfaces configured on the following hardware only:</p> <ul style="list-style-type: none"> • Adaptive Services (AS) PICs on M Series and T Series routers • Multiservices (MS) PICs on M Series and T Series routers • Multiservices (MS) DPCs on MX Series routers

Table 2: Filter Types (*continued*)

Filter Type	Filter Configuration Statement	Description
Simple Filter	simple-filter <i>simple-filter-name</i>	<p>Defines packet filtering to be applied to ingress traffic only.</p> <p>Filters the following traffic type:</p> <ul style="list-style-type: none"> • IPv4 <p>Supported at logical interfaces configured on the following hardware only:</p> <ul style="list-style-type: none"> • Gigabit Ethernet Intelligent Queuing (IQ2) PICs installed on M120, M320, or T Series routers • Enhanced Queuing Dense Port Concentrators (EQ DPCs) installed on MX Series routers

Terms

Under the **filter**, **service-filter**, or **simple-filter** statement, you must configure at least one firewall filter *term*. A term is a named structure in which match conditions and actions are defined. Within a firewall filter, you must configure a unique name for each term.



TIP: You cannot apply a different filter on each direction of traffic on the same interface. As a result, it is common to create firewall filters with multiple terms.

If a packet arrives on an interface for which no firewall filter is applied for the incoming traffic on that interface, the packet is accepted by default.

Match Conditions

A firewall filter term must contain at least one packet-filtering criteria, called a *match condition*, to specify the field or value that a packet must contain in order to be considered a match for the firewall filter term. For a match to occur, the packet must match all the conditions in the term. If a packet matches a firewall filter term, the router takes the configured action on the packet.

If a firewall filter term contains multiple match conditions, a packet must meet *all* match conditions to be considered a match for the firewall filter term.

If a single match condition is configured with multiple values, such as a range of values, a packet must match only *one* of the values to be considered a match for the firewall filter term.

The scope of match conditions you can specify in a firewall filter term depends on the protocol family under which the firewall filter is configured. You can define various match conditions, including the IP source address field, IP destination address field, TCP or UDP source port field, IP protocol field, Internet Control Message Protocol (ICMP) packet type, IP options, TCP flags, incoming logical or physical interface, and outgoing logical or physical interface.

Each protocol family supports a different set of match conditions, and some match conditions are supported only on certain routing devices. For example, a number of match conditions for VPLS traffic are supported only on the MX Series 3D Universal Edge Routers.

In the **from** statement in a firewall filter term, you specify characteristics that the packet must have for the action in the subsequent **then** statement to be performed. The characteristics are referred to as *match conditions*. The packet must match all conditions in the **from** statement for the action to be performed, which also means that the order of the conditions in the **from** statement is not important.

If an individual match condition can specify a list of values (such as multiple source and destination addresses) or a range of numeric values, a match occurs if any of the values matches the packet.

If a filter term does not specify match conditions, the term accepts all packets and the actions specified in the term's **then** statement are optional.

Actions

The actions specified in a firewall filter term define the actions to take for any packet that matches the conditions specified in the term.

Actions that are configured within a single term are all taken on traffic that matches the conditions configured.



BEST PRACTICE: We strongly recommend that you explicitly configure one or more actions per firewall filter term. Any packet that matches all the conditions of the term is automatically accepted unless the term specifies other or additional actions.

Firewall filter actions fall into the following categories:

Filter-Terminating Actions

A filter-terminating action halts all evaluation of a firewall filter for a specific packet. The router performs the specified action, and no additional terms are examined.

Nonterminating Actions

Nonterminating actions are used to perform other functions on a packet, such as incrementing a counter, logging information about the packet header, sampling the packet data, or sending information to a remote host using the system log functionality.

Flow Control Action

For standard stateless firewall filters only, the action **next term** enables the router to perform configured actions on the packet and then evaluate the following term in the filter, rather than terminating the filter.

A maximum of 1024 **next term** actions are supported per standard stateless firewall filter configuration. If you configure a standard filter that exceeds this limit, your candidate configuration results in a commit error.

- Related Documentation**
- Stateless Firewall Filter Types on page 6
 - Stateless Firewall Filter Application Points on page 11
 - “Inserting a New Identifier in a Junos Configuration” in the *Junos OS CLI User Guide*

Stateless Firewall Filter Application Points

After you define the firewall filter, you must apply it to an application point. These application points include logical interfaces, physical interfaces, routing interfaces, and routing instances.

In most cases, you can apply a firewall filter as an *input* filter or an *output* filter, or both at the same time. Input filters take action on packets being received on the specified interface, whereas output filters take action on packets that are transmitted through the specified interface.

You typically apply one filter with multiple terms to a single logical interface, to incoming traffic, outbound traffic, or both. However, there are times when you might want to chain together multiple firewall filters (with single or multiple terms) and apply them to an interface. You use an *input list* to apply multiple firewall filters to the incoming traffic on an interface. You use an *output list* to apply multiple firewall filters to the outbound traffic on an interface. You can include up to 16 filters in an input list or an output list.

There is no limit to the number of filters and counters you can set, but there are some practical considerations. More counters require more terms, and a large number of terms can take a long time to process during a commit operation. However, filters with more than 4000 terms and counters have been implemented successfully.

Table 3 on page 11 describes each point to which you can apply a firewall filter. For each application point, the table describes the types of firewall filters supported at that point, the router hierarchy level at which the filter can be applied, and any platform-specific limitations.

Table 3: Stateless Firewall Filter Configuration and Application Summary

Filter Type	Application Point	Restrictions
Stateless firewall filter Configure by including the filter <i>filter-name</i> statement the [edit firewall] hierarchy level: <pre>filter filter-name;</pre> <p>NOTE: If you do not include the family statement, the firewall filter processes IPv4 traffic by default.</p>	Logical interface Apply at the [edit interfaces <i>interface-name</i> unit <i>unit-number</i> family inet] hierarchy level by including the input <i>filter-name</i> or output <i>filter-name</i> statements: <pre>filter { input filter-name; output filter-name; }</pre> <p>NOTE: A filter configured with the implicit inet protocol family cannot be included in an input filter list or an output filter list.</p>	Supported on the following routers: <ul style="list-style-type: none"> • T Series routers • M320 routers • M7i routers with the enhanced CFEB (CFEB-e) • M10i routers with the enhanced CFEB-e Also supported on the following Modular Port Concentrators (MPCs) on MX Series routers: <ul style="list-style-type: none"> • 10-Gigabit Ethernet MPC • 60-Gigabit Ethernet Queuing MPC • 60-Gigabit Ethernet Enhanced Queuing MPC

Table 3: Stateless Firewall Filter Configuration and Application Summary (*continued*)

Filter Type	Application Point	Restrictions
<p>Stateless firewall filter</p> <p>Configure at the [edit firewall family <i>family-name</i>] hierarchy level by including the following statement:</p> <pre>filter <i>filter-name</i>;</pre> <p>The <i>family-name</i> can be any of the following protocol families:</p> <ul style="list-style-type: none"> • any • bridge • ccc • inet • inet6 • mpls • vpls 	<p>Protocol family on a logical interface</p> <p>Apply at the [edit interfaces <i>interface-name</i> unit <i>unit-number</i> family <i>family-name</i>] hierarchy level by, including the input, input-list, output, or output-list statements:</p> <pre>filter { input <i>filter-name</i>; input-list [<i>filter-names</i>]; output <i>filter-name</i>; output-list [<i>filter-names</i>]; }</pre>	<p>The protocol family bridge is supported only on MX Series routers.</p>
<p>Stateless firewall filter</p>	<p>Routing Engine loopback interface</p>	
<p>Service filter</p> <p>Configure at the [edit firewall family (inet inet6)] hierarchy level by including the following statement:</p> <pre>service-filter <i>service-filter-name</i>;</pre>	<p>Family inet or inet6 on a logical interface</p> <p>Apply at the [edit interfaces <i>interface-name</i> unit <i>unit-number</i> family (inet inet6)] hierarchy level by using the service-set statement to apply a service filter as an input or output filter to a service set:</p> <pre>service { input { service-set <i>service-set-name</i> service-filter <i>filter-name</i>; } output { service-set <i>service-set-name</i> service-filter <i>filter-name</i>; } }</pre> <p>Configure a service set at the [edit services] hierarchy level by including the following statement:</p> <pre>service-set <i>service-set-name</i>;</pre>	<p>Supported only on Adaptive Services (AS) and Multiservices (MS) PICs.</p>

Table 3: Stateless Firewall Filter Configuration and Application Summary (*continued*)

Filter Type	Application Point	Restrictions
Postservice filter Configure at the [edit firewall family (inet inet6)] hierarchy level by including the following statement: <pre>service-filter service-filter-name;</pre>	Family inet or inet6 on a logical interface Apply at the [edit interfaces interface-name unit unit-number family (inet inet6)] hierarchy level by including the post-service-filter statement to apply a service filter as an input filter: <pre>service { input { post-service-filter filter-name; } }</pre>	A postservice filter is applied to traffic returning to the services interface after service processing. The filter is applied only if a service set is configured and selected.
Simple filter Configure at the [edit firewall family inet] hierarchy level by including the following statement: <pre>simple-filter filter-name</pre>	Family inet on a logical interface Apply at the [edit interfaces interface-name unit unit-number family inet] hierarchy level by including the following statement: <pre>simple-filter simple-filter-name;</pre>	Simple filters can only be applied as input filters. Supported on the following platforms only: <ul style="list-style-type: none"> Gigabit Ethernet intelligent queuing (IQ2) PICs on the M120, M320, and T Series routers. Enhanced Queuing Dense Port Concentrators (EQ DPC) on MX Series routers.
Reverse packet forwarding (RPF) check filter Configured at the [edit firewall family (inet inet6)] hierarchy level by including the following statement: <pre>filter filter-name;</pre>	Family inet or inet6 on a logical interface Apply at the [edit interfaces interface-name unit unit-number family (inet inet6)] hierarchy level by including the following statement: <pre>rpf-check fail-filter filter-name</pre> to apply the stateless firewall filter as an RPF check filter. <pre>rpf-check { fail-filter filter-name; mode loose; }</pre>	Supported on MX Series routers only.

- Related Documentation**
- Stateless Firewall Filter Components on page 7
 - Supported Standards for Filtering on page 229

CHAPTER 2

Standard Firewall Filter Overview

- [Standard Stateless Firewall Filter Overview on page 15](#)
- [How Standard Firewall Filters Evaluate Packets on page 16](#)
- [Guidelines for Configuring Standard Firewall Filters on page 17](#)
- [Guidelines for Applying Standard Firewall Filters on page 22](#)
- [Basic Uses for Standard Firewall Filters on page 24](#)

Standard Stateless Firewall Filter Overview

Firewall filters provide a means of protecting your router from excessive traffic transiting the router to a network destination or destined for the Routing Engine. Firewall filters that control local packets can also protect your router from external incidents.

You can configure a firewall filter to do the following:

- Restrict traffic destined for the Routing Engine based on its source, protocol, and application.
- Limit the traffic rate of packets destined for the Routing Engine to protect against flood, or denial-of-service (DoS) attacks.
- Address special circumstances associated with fragmented packets destined for the Routing Engine. Because the device evaluates every packet against a firewall filter (including fragments), you must configure the filter to accommodate fragments that do not contain packet header information. Otherwise, the filter discards all but the first fragment of a fragmented packet.

Related Documentation

- [Stateless Firewall Filter Types on page 6](#)
- [How Standard Firewall Filters Evaluate Packets on page 16](#)
- [Guidelines for Configuring Standard Firewall Filters on page 17](#)
- [Guidelines for Applying Standard Firewall Filters on page 22](#)
- [Basic Uses for Standard Firewall Filters on page 24](#)

How Standard Firewall Filters Evaluate Packets

This topic covers the following information:

- Firewall Filters That Contain a Single Term on page 16
- Firewall Filters That Contain Multiple Terms on page 16
- Firewall Filter Terms That Do Not Contain Any Match Conditions on page 17
- Firewall Filter Terms That Do Not Contain Any Actions on page 17
- Firewall Filter Default Action on page 17

Firewall Filters That Contain a Single Term

For a standard stateless firewall filter that consists of a single term, the policy framework software evaluates a packet as follows:

- If the packet matches all the conditions, the actions are taken.
- If the packet matches all the conditions and no actions are specified, the packet is accepted.
- If the packet does not match all the conditions, it is discarded.

Firewall Filters That Contain Multiple Terms

For a standard stateless firewall filter that consists of multiple terms, the policy framework software evaluates a packet against the terms in the filter sequentially, beginning with the first term in the filter, until either the packet matches all the conditions in one of the terms or there are no more terms in the filter.

- If the packet matches all the conditions in a term, the actions in that term are performed.
- Unlike service filters and simple filters, standard stateless firewall filters support the **next term** action, which is neither a terminating action nor a nonterminating action but a flow control action.:
 - If the actions do not include the **next term** action, evaluation of the packet ends at this term of the filter, and any subsequent terms in the filter are not used.
 - If the actions include the **next term** action, the evaluation continues to the next term.
- If the packet does not match all the conditions in the term, evaluation of the packet proceeds to the next term in the filter.
- Evaluation of the packet continues until either the packet matches a term without **next term** action or until the end of the filter is reached.

A maximum of 1024 **next term** actions are supported per standard stateless firewall filter configuration. If you configure a standard filter that exceeds this limit, your candidate configuration results in a commit error.

Firewall Filter Terms That Do Not Contain Any Match Conditions

For standard stateless filters with a single term and for standard stateless firewall filters with multiple terms, if a term does not specify any match conditions, the actions are taken on any packet evaluated.

Firewall Filter Terms That Do Not Contain Any Actions

If a standard stateless firewall filter term does not contain any actions, and if the packet matches the conditions in the term, the packet is accepted.

Firewall Filter Default Action

Each standard stateless firewall filter has an *implicit* **discard** action at the end of the filter, which is equivalent to including the following example term **explicit_discard** as the final term in the standard stateless firewall filter:

```
term explicit_discard {  
    then discard;  
}
```

By default, if a packet matches none of the terms in a standard stateless firewall filter, the packet is discarded.

**Related
Documentation**

- How Service Filters Evaluate Packets on page 62
- How Simple Filters Evaluate Packets on page 69
- Guidelines for Configuring Standard Firewall Filters on page 17
- Basic Uses for Standard Firewall Filters on page 24

Guidelines for Configuring Standard Firewall Filters

This topic covers the following information:

- Statement Hierarchy for Configuring Standard Firewall Filters on page 17
- Standard Firewall Filter Protocol Families on page 18
- Standard Firewall Filter Names and Options on page 19
- Standard Firewall Filter Terms on page 19
- Standard Firewall Filter Match Conditions on page 19
- Standard Firewall Filter Actions on page 21

Statement Hierarchy for Configuring Standard Firewall Filters

To configure a standard firewall filter, you can include the following statements. For an IPv4 standard firewall filter, the **family inet** statement is optional.

```
firewall {  
    family family-name {  
        filter filter-name {  
            accounting-profile name;  
            interface-specific;  
            physical-interface-filter;  
            term term-name {
```

```

    filter filter-name;
  }
  term term-name {
    from {
      match-conditions;
      ip-version ipv4 {
        match-conditions;
        protocol (tcp | udp) {
          match conditions;
        }
      }
    }
    then {
      actions;
    }
  }
}

```

You can include the firewall configuration at one of the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**



NOTE: For stateless firewall filtering, you must allow the output tunnel traffic through the firewall filter applied to input traffic on the interface that is the next-hop interface toward the tunnel destination. The firewall filter affects only the packets exiting the router by way of the tunnel.

Standard Firewall Filter Protocol Families

A standard firewall filter configuration is specific to a particular protocol family. Under the **firewall** statement, include one of the following statements to specify the protocol family for which you want to filter traffic:

- **family any**—To filter protocol-independent traffic.
- **family inet**—To filter Internet Protocol version 4 (IPv4) traffic.
- **family inet6**—To filter Internet Protocol version 6 (IPv6) traffic.
- **family mpls**—To filter MPLS traffic.
- **family vpls**—To filter virtual private LAN service (VPLS) traffic.
- **family ccc**—To filter Layer 2 circuit cross-connection (CCC) traffic.
- **family bridge**—To filter Layer 2 bridging traffic for MX Series 3D Universal Edge Routers only.

The **family *family-name*** statement is required only to specify a protocol family other than IPv4. To configure an IPv4 firewall filter, you can configure the filter at the **[edit firewall]**

hierarchy level without including the **family inet** statement, because the **[edit firewall]** and **[edit firewall family inet]** hierarchy levels are equivalent.

Standard Firewall Filter Names and Options

Under the **family *family-name*** statement, you can include **filter *filter-name*** statements to create and name standard firewall filters. The filter name can contain letters, numbers, and hyphens (-) and be up to 64 characters long. To include spaces in the name, enclose the entire name in quotation marks (" ").

At the **[edit firewall family *family-name* filter *filter-name*]** hierarchy level, the following statements are optional:

- **accounting-profile**
- **interface-specific**
- **physical-interface-filter**

Standard Firewall Filter Terms

Under the **filter *filter-name*** statement, you can include **term *term-name*** statements to create and name filter terms.

- You must configure at least one term in a firewall filter.
- You must specify a unique name for each term within a firewall filter. The term name can contain letters, numbers, and hyphens (-) and can be up to 64 characters long. To include spaces in the name, enclose the entire name in quotation marks (" ").
- The order in which you specify terms within a firewall filter configuration is important. Firewall filter terms are evaluated in the order in which they are configured. By default, new terms are always added to the end of the existing filter. You can use the **insert** configuration mode command to reorder the terms of a firewall filter.

At the **[edit firewall family *family-name* filter *filter-name* term *term-name*]** hierarchy level, the **filter *filter-name*** statement is not valid in the same term as **from** or **then** statements. When included at this hierarchy level, the **filter *filter-name*** statement is used to *nest* firewall filters.

Standard Firewall Filter Match Conditions

Standard firewall filter match conditions are specific to the type of traffic being filtered.

With the exception of MPLS-tagged IPv4 traffic, you specify the term's match conditions under the **from** statement. For MPLS-tagged IPv4 traffic, you specify the term's IPv4 address-specific match conditions under the **ip-version *ipv4*** statement and the term's IPv4 port-specific match conditions under the **protocol (tcp | udp)** statement.

Table 4 on page 20 describes the types of traffic for which you can configure standard stateless firewall filters.

Table 4: Standard Firewall Filter Match Conditions by Protocol Family

Traffic Type	Hierarchy Level at Which Match Conditions Are Specified
Protocol-independent	<p>[edit firewall family any filter <i>filter-name</i> term <i>term-name</i>]</p> <p>For the complete list of match conditions, see “Standard Firewall Filter Match Conditions for Protocol-Independent Traffic” on page 235.</p>
IPv4	<p>[edit firewall family inet filter <i>filter-name</i> term <i>term-name</i>]</p> <p>For the complete list of match conditions, see “Standard Firewall Filter Match Conditions for IPv4 Traffic” on page 236.</p>
IPv6	<p>[edit firewall family inet6 filter <i>filter-name</i> term <i>term-name</i>]</p> <p>For the complete list of match conditions, see “Standard Firewall Filter Match Conditions for IPv6 Traffic” on page 245.</p>
MPLS	<p>[edit firewall family mpls filter <i>filter-name</i> term <i>term-name</i>]</p> <p>For the complete list of match conditions, see “Standard Firewall Filter Match Conditions for MPLS Traffic” on page 250.</p>
IPv4 addresses in MPLS flows	<p>[edit firewall family mpls filter <i>filter-name</i> term <i>term-name</i> ip-version ipv4]</p> <p>For the complete list of match conditions, see “Standard Firewall Filter Match Conditions for MPLS-Tagged IPv4 Traffic” on page 252.</p>
IPv4 ports in MPLS flows	<p>[edit firewall family mpls filter <i>filter-name</i> term <i>term-name</i> ip-version ipv4 protocol (tcp udp)]</p> <p>For the complete list of match conditions, see “Standard Firewall Filter Match Conditions for MPLS-Tagged IPv4 Traffic” on page 252.</p>
VPLS	<p>[edit firewall family vpls filter <i>filter-name</i> term <i>term-name</i>]</p> <p>For the complete list of match conditions, see “Standard Firewall Filter Match Conditions for VPLS Traffic” on page 254.</p>
Layer 2 CCC	<p>[edit firewall family ccc filter <i>filter-name</i> term <i>term-name</i>]</p> <p>For the complete list of match conditions, see “Standard Firewall Filter Match Conditions for Layer 2 CCC Traffic” on page 261.</p>
Layer 2 Bridging	<p>[edit firewall family bridge filter <i>filter-name</i> term <i>term-name</i>]</p>
(MX Series routers only)	<p>For the complete list of match conditions, see “Standard Firewall Filter Match Conditions for Layer 2 Bridging Traffic” on page 263.</p>

If you specify an IPv6 address in a match condition (the **address**, **destination-address**, or **source-address** match conditions), use the syntax for text representations described in RFC 2373, *IP Version 6 Addressing Architecture*. For more information about IPv6 addresses, see “IPv6 Overview” and “IPv6 Standards” in the [Junos OS Routing Protocols Configuration Guide](#).

Standard Firewall Filter Actions

Under the **then** statement for a standard stateless firewall filter term, you can specify the actions to be taken on a packet that matches the term.

Table 5 on page 21 summarizes the types of actions you can specify in a standard stateless firewall filter term.

Table 5: Standard Firewall Filter Action Categories

Type of Action	Description	Comment
Terminating	<p>Halts all evaluation of a firewall filter for a specific packet. The router performs the specified action, and no additional terms are used to examine the packet.</p> <p>You can specify only one <i>terminating action</i> in a standard firewall filter. You can, however, specify one terminating action with one or more <i>nonterminating actions</i> in a single term. For example, within a term, you can specify accept with count and syslog.</p>	See “Standard Firewall Filter Terminating Actions” on page 269.
Nonterminating	<p>Performs other functions on a packet (such as incrementing a counter, logging information about the packet header, sampling the packet data, or sending information to a remote host using the system log functionality), but any additional terms are used to examine the packet.</p>	See “Standard Firewall Filter Nonterminating Actions” on page 270.
Flow control	<p>For standard stateless firewall filters only, the next term action directs the router to perform configured actions on the packet and then, rather than terminate the filter, use the next term in the filter to evaluate the packet. If the next term action is included, the matching packet is evaluated against the next term in the firewall filter. Otherwise, the matching packet is not evaluated against subsequent terms in the firewall filter.</p> <p>For example, when you configure a term with the nonterminating action count, the term’s action changes from an implicit discard to an implicit accept. The next term action forces the continued evaluation of the firewall filter.</p>	<p>You cannot configure the next term action with a terminating action in the same filter term. However, you can configure the next term action with another nonterminating action in the same filter term.</p> <p>A maximum of 1024 next term actions are supported per standard stateless firewall filter configuration. If you configure a standard filter that exceeds this limit, your candidate configuration results in a commit error.</p>

- Related Documentation**
- How Standard Firewall Filters Evaluate Packets on page 16
 - Guidelines for Applying Standard Firewall Filters on page 22

Guidelines for Applying Standard Firewall Filters

This topic covers the following information:

- Applying Standard Firewall Filters Overview on page 22
- Statement Hierarchy for Applying Standard Firewall Filters on page 22
- Restrictions on Applying Standard Firewall Filters on page 23

Applying Standard Firewall Filters Overview

You can apply a standard stateless firewall filter to a physical interface on the router or to the loopback interface on the router. You can apply a firewall filter to a single interface or to multiple interfaces on the router.

Applying a Firewall Filter to a Router's Physical Interfaces

When you apply a standard firewall filter to a physical interface on the router, the filter evaluates all data packet that pass through that interface.

Applying a Firewall Filter to the Router's Loopback Interface

The router's loopback interface, **lo0**, is the interface to the Routing Engine and carries no data packets. When you apply a standard firewall filter to the loopback interface, the filter evaluates the local packets received or transmitted by the Routing Engine.

Applying a Firewall Filter to Multiple Interfaces

You can use the same standard firewall filter one or more times.

On M Series routers, except the M120 and M320 routers, if you apply a firewall filter to multiple interfaces, the filter acts on the sum of traffic entering or exiting those interfaces.

On T Series, M120, M320, and MX Series routers, interfaces are distributed among multiple packet-forwarding components. On these routers, you can configure standard stateless firewall filters and service filters that, when applied to multiple interfaces, act on the individual traffic streams entering or exiting each interface, regardless of the sum of traffic on the multiple interfaces.

For more information, see "Interface-Specific Firewall Filter Instances Overview" on page 51.

Statement Hierarchy for Applying Standard Firewall Filters

To apply a standard stateless firewall filter to a logical interface, configure the **input filter-name**, **input-list filter-name**, **output filter-name**, or **output-list filter-name** statements in the **filter** stanza for the logical interface protocol family.

```
interfaces {  
  interface-name {  
    unit logical-unit-number {  
      family family-name {  
        ...  
      }  
    }  
  }  
}
```



```

filter {
  group group-number;
  input filter-name;
  input-list [ filter-names ];
  output filter-name;
  output-list [ filter-names ];
}
}
}
}
}

```

You can include the interface configuration at one of the following hierarchy levels:

- [edit]
- [edit logical-systems *logical-system-name*]

Restrictions on Applying Standard Firewall Filters

- Number of Input and Output Filters Per Logical Interface on page 23
- MPLS and Layer 2 CCC Firewall Filters in Lists on page 23
- Layer 2 CCC Firewall Filters on MX Series Routers on page 24
- Protocol-Independent Firewall Filters on the Loopback Interface on page 24

Number of Input and Output Filters Per Logical Interface

Input filters—Although you can use the same filter multiple times, you can apply only one input filter or one input filter list to an interface.

- To specify a single firewall filter to be used to evaluate packets received on the interface, include the **input *filter-name*** statement in the **filter** stanza.
- To specify an ordered list of firewall filters to be used to evaluate packets received on the interface, include the **input-list [*filter-names*]** statement in the **filter** stanza. You can specify up to 16 firewall filters for the filter input list.

Output filters—Although you can use the same filter multiple times, you can apply only one output filter or one output filter list to an interface.

- To specify a single firewall filter to be used to evaluate packets transmitted on the interface, include the **output *filter-name*** statement in the **filter** stanza.
- To specify an ordered list of firewall filters to be used to evaluate packets transmitted on the interface, include the **output-list [*filter-names*]** statement in the **filter** stanza. You can specify up to 16 firewall filters in a filter output list.

MPLS and Layer 2 CCC Firewall Filters in Lists

The **input-list *filter-names*** and **output-list *filter-names*** statements for firewall filters for the **ccc** and **mpls** protocol families are supported on all interfaces with the exception of the following:

- Management interfaces and internal Ethernet interfaces (**fxp** or **em0**)
- Loopback interfaces (**lo0**)

- USB modem interfaces (umd)

Layer 2 CCC Firewall Filters on MX Series Routers

On MX Series routers only, you cannot apply a Layer 2 CCC stateless firewall filter (a firewall filter configured at the **[edit firewall filter family ccc]** hierarchy level) as an output filter. On MX Series routers, firewall filters configured for the **family ccc** statement can be applied only as input filters.

Protocol-Independent Firewall Filters on the Loopback Interface

Protocol-independent firewall filters—stateless firewall filters configured at the **[edit firewall family any]** hierarchy level—are not supported on the router loopback interface (lo0).

Related Documentation

- How Standard Firewall Filters Evaluate Packets on page 16
- Guidelines for Configuring Standard Firewall Filters on page 17
- Basic Uses for Standard Firewall Filters on page 24

Basic Uses for Standard Firewall Filters

This topic covers the following information:

- Using Standard Firewall Filters to Affect Local Packets on page 24
- Using Standard Firewall Filters to Affect Data Packets on page 25

Using Standard Firewall Filters to Affect Local Packets

On a router, you can configure one physical loopback interface, **lo0**, and one or more addresses on the interface. The loopback interface is the interface to the Routing Engine, which runs and monitors all the control protocols. The loopback interface carries local packets only. Standard firewall filters applied to the loopback interface affect the local packets destined for or transmitted from the Routing Engine.



NOTE: When you create an additional loopback interface, it is important to apply a filter to it so the Routing Engine is protected. We recommend that when you apply a filter to the loopback interface, you include the **apply-groups** statement. Doing so ensures that the filter is automatically inherited on every loopback interface, including **lo0** and other loopback interfaces.

Trusted Sources

The typical use of a standard stateless firewall filter is to protect the Routing Engine processes and resources from malicious or untrusted packets. To protect the processes and resources owned by the Routing Engine, you can use a standard stateless firewall filter that specifies which protocols and services, or applications, are allowed to reach the Routing Engine. Applying this type of filter to the loopback interface ensures that the local packets are from a trusted source and protects the processes running on the Routing Engine from an external attack.

Flood Prevention

You can create standard stateless firewall filters that limit certain TCP and ICMP traffic destined for the Routing Engine. A router without this kind of protection is vulnerable to TCP and ICMP flood attacks, which are also called denial-of-service (DoS) attacks. For example:

- A TCP flood attack of SYN packets initiating connection requests can overwhelm the device until it can no longer process legitimate connection requests, resulting in denial of service.
- An ICMP flood can overload the device with so many echo requests (ping requests) that it expends all its resources responding and can no longer process valid network traffic, also resulting in denial of service.

Applying the appropriate firewall filters to the Routing Engine protects against these types of attacks.

Using Standard Firewall Filters to Affect Data Packets

Standard firewall filters that you apply to your router's transit interfaces evaluate only the user data packets that transit the router from one interface directly to another as they are being forwarded from a source to a destination. To protect the network as a whole from unauthorized access and other threats at specific interfaces, you can apply firewall filters router transit interfaces .

**Related
Documentation**

- [How Standard Firewall Filters Evaluate Packets on page 16](#)
- [Guidelines for Configuring Standard Firewall Filters on page 17](#)
- [Guidelines for Applying Standard Firewall Filters on page 22](#)

CHAPTER 3

Standard Firewall Filter Match Conditions Overview

- Firewall Filter Match Conditions Based on Numbers or Text Aliases on page 27
- Firewall Filter Match Conditions Based on Bit-Field Values on page 28
- Firewall Filter Match Conditions Based on Address Fields on page 32
- Firewall Filter Match Conditions Based on Address Classes on page 39

Firewall Filter Match Conditions Based on Numbers or Text Aliases

This topic covers the following information:

- Matching on a Single Numeric Value on page 27
- Matching on a Range of Numeric Values on page 27
- Matching on a Text Alias for a Numeric Value on page 28
- Matching on a List of Numeric Values or Text Aliases on page 28

Matching on a Single Numeric Value

You can specify a firewall filter match condition based on whether a particular packet field value is a specified numeric value. In the following example, a match occurs if the packet source port number is **25**:

```
[edit firewall family inet filter filter1 term term1 from]
user@host# set source-port 25
```

Matching on a Range of Numeric Values

You can specify a firewall filter match condition based on whether a particular packet field value falls within a specified range of numeric values. In the following example, a match occurs for source ports values from **1024** through **65,535**, inclusive:

```
[edit firewall family inet filter filter2 term term1 from]
user@host# set source-port 1024-65535
```

Matching on a Text Alias for a Numeric Value

You can specify a firewall filter match condition based on whether a particular packet field value is a numeric value that you specify by using a text string as an *alias* for the numeric value. In the following example, a match occurs if the packet source port number is 25. For the **source-port** and **destination-port** match conditions, the text alias **smtp** corresponds to the numeric value 25.

```
[edit firewall family inet filter filter3 term term1 from]
user@host# set source-port smtp
```

Matching on a List of Numeric Values or Text Aliases

You can specify a firewall filter match condition based on whether a particular packet field value matches any one of multiple numeric values or text aliases that you specify within square brackets and delimited by spaces. In the following example, a match occurs if the packet source port number is any of the following values: 20 (which corresponds to the text aliases **ftp-data**), 25, or any value from 1024 through 65535.

```
[edit firewall family inet filter filter3 term term1 from]
user@host# set source-port [ smtp ftp-data 25 1024-65535 ]
```

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 17
- Firewall Filter Match Conditions Based on Bit-Field Values on page 28
- Firewall Filter Match Conditions Based on Address Fields on page 32
- Firewall Filter Match Conditions Based on Address Classes on page 39

Firewall Filter Match Conditions Based on Bit-Field Values

- Match Conditions for Bit-Field Values on page 28
- Match Conditions for Common Bit-Field Values or Combinations on page 29
- Logical Operators for Bit-Field Values on page 30
- Matching on a Single Bit-Field Value or Text Alias on page 30
- Matching on Multiple Bit-Field Values or Text Aliases on page 31
- Matching on a Negated Bit-Field Value on page 31
- Matching on the Logical OR of Two Bit-Field Values on page 31
- Matching on the Logical AND of Two Bit-Field Values on page 32
- Grouping Bit-Field Match Conditions on page 32

Match Conditions for Bit-Field Values

Table 6 on page 29 lists the firewall filter match conditions that are based on whether certain bit fields in a packet are set or not set. The second and third columns list the types of traffic for which the match condition is supported.

Table 6: Binary and Bit-Field Match Conditions for Firewall Filters

Bit-Field Match Condition	Match Values	Protocol Families for Standard Stateless Firewall Filters	Protocol Families for Service Filters
fragment-flags <i>flags</i>	Hexadecimal values or text aliases for the three-bit IP fragmentation flags field in the IP header.	family inet	family inet
fragment-offset <i>value</i>	Hexadecimal values or text aliases for the 13-bit fragment offset field in the IP header.	family inet	family inet
tcp-flags <i>value</i> [†]	Hexadecimal values or text aliases for the low-order 6 bits of the 8-bit TCP flags field in the TCP header.	family inet family inet6 family vpls family bridge	family inet family inet6

[†] The Junos OS does not automatically check the first fragment bit when matching TCP flags for IPv4 traffic. To check the first fragment bit for IPv4 traffic only, use the **first-fragment** match condition.

Match Conditions for Common Bit-Field Values or Combinations

Table 7 on page 29 describes firewall filter match conditions that are based on whether certain commonly used values or *combinations* of bit fields in a packet are set or not set.

Table 7: Bit-Field Match Conditions for Common Combinations

Match Condition	Description	Protocol Families for Standard Stateless Firewall Filters	Protocol Families for Service Filters
first-fragment	Text alias for the bit-field match condition fragment-offset 0 , which indicates the first fragment of a fragmented packet.	family inet	family inet
is-fragment	Text alias for the bit-field match condition fragment-offset 0 except , which indicates a trailing fragment of a fragmented packet.	family inet	family inet
tcp-established	Alias for the bit-field match condition tcp-flags "(ack rst)" , which indicates an established TCP session, but not the first packet of a TCP connection.	family inet family inet6	—

Table 7: Bit-Field Match Conditions for Common Combinations (*continued*)

Match Condition	Description	Protocol Families for Standard Stateless Firewall Filters	Protocol Families for Service Filters
tcp-initial	Alias for the bit-field match condition tcp-flags "(!ack & syn)" , which indicates the first packet of a TCP connection, but not an established TCP session.	family inet family inet6	—

Logical Operators for Bit-Field Values

Table 8 on page 30 lists the logical operators you can apply to *single* bit-field values when specifying stateless firewall filter match conditions. The operators are listed in order, from highest precedence to lowest precedence. Operations are left-associative, meaning that the operations are processed from left to right.

Table 8: Bit-Field Logical Operators

Precedence Order	Bit-Field Logical Operator	Description
1	(complex-match-condition)	Grouping—The complex match condition is evaluated before any operators outside the parentheses are applied.
2	! match-condition	Negation—A match occurs if the match condition is false.
3	match-condition-1 & match-condition-2 or match-condition-1 + match-condition-2	Logical AND—A match occurs if both match conditions are true.
4	match-condition-1 match-condition-2 or match-condition-1 , match-condition-2	Logical OR—A match occurs if either match condition is true.

Matching on a Single Bit-Field Value or Text Alias

For the **fragment-flags** and **tcp-flags** bit-match conditions, you can specify firewall filter match conditions based on whether a particular bit in the packet field is set or not set.

- Numeric value to specify a single bit—You can specify a single bit-field match condition by using a numeric value that has one bit set. Depending on the match condition, you can specify a decimal value, a binary value, or a hexadecimal value. To specify a binary value, specify the number with the prefix **b**. To specify a hexadecimal value, specify the number with the prefix **0x**.

In the following example, a match occurs if the **RST** bit in the TCP flags field is set:

```
[edit firewall family inet filter filter_tcp_rst_number term term1 from]
user@host# set tcp-flags 0x04
```


- Text alias to specify a single bit—You generally specify a single bit-field match condition by using a text alias enclosed in double-quotation marks (“ ”).

In the following example, a match occurs if the **RST** bit in the TCP flags field is set:

```
[edit firewall family inet filter filter_tcp_rst_alias term term1 from]
user@host# set tcp-flags "rst"
```

Matching on Multiple Bit-Field Values or Text Aliases

You can specify a firewall filter match condition based on whether a particular set of bits in a packet field are set.

- Numeric values to specify multiple set bits—When you specify a numeric value whose binary representation has more than one set bit, the value is treated as a logical AND of the set bits.

In the following example, the two match conditions are the same. A match occurs if either bit **0x01** or **0x02** is not set:

```
[edit firewall family inet filter reset_or_not_initial_packet term term5 from]
user@host# set tcp-flags "!0x3"
user@host# set tcp-flags "!(0x01 & 0x02)"
```

- Text aliases that specify common bit-field matches—You can use text aliases to specify some common bit-field matches. You specify these matches as a single keyword.

In the following example, the **tcp-established** condition, which is an alias for “(**ack** | **rst**)”, specifies that a match occurs on TCP packets other than the first packet of a connection:

```
[edit firewall family inet filter reset_or_not_initial_packet term term6 from]
user@host# set tcp-established
```

Matching on a Negated Bit-Field Value

To negate a match, precede the value with an exclamation point.

In the following example, a match occurs if the **RST** bit in the TCP flags field is *not* set:

```
[edit firewall family inet filter filter_tcp_rst term term1 from]
user@host# set tcp-flags "!rst"
```

Matching on the Logical OR of Two Bit-Field Values

You can use the *logical OR operator* (| or ,) to specify that a match occurs if a bit field matches either of two bit-field values specified.

In the following example, a match occurs if the packet is *not* the initial packet in a TCP session:

```
[edit firewall family inet filter not_initial_packet term term3 from]
user@host# set tcp-flags "!syn | ack"
```

In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet. In a packet that is not the initial packet in a TCP session, either the SYN flag is not set or the ACK flag is set.

Matching on the Logical AND of Two Bit-Field Values

You can use the *logical AND operator* (& or +) to specify that a match occurs if a bit field matches both of two bit-field values specified.

In the following example, a match occurs if the packet is the initial packet in a TCP session:

```
[edit firewall family inet filter initial_packet term term2 from]
user@host# set tcp-flags "syn & !ack"
```

In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet. In a packet that is an initial packet in a TCP session, the SYN flag is set and the ACK flag is not set.

Grouping Bit-Field Match Conditions

You can use the *logical grouping notation* to specify that the complex match condition inside the parentheses is evaluated before any operators outside the parentheses are applied.

In the following example, a match occurs if the packet is a TCP reset or if the packet is not the initial packet in the TCP session:

```
[edit firewall family inet filter reset_or_not_initial_packet term term4 from]
user@host# set tcp-flags "!(syn & !ack) | rst"
```

In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet. In a packet that is *not* the initial packet in a TCP session, the SYN flag is not set and the ACK field is set.

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 17
- Firewall Filter Match Conditions Based on Numbers or Text Aliases on page 27
- Firewall Filter Match Conditions Based on Address Fields on page 32
- Firewall Filter Match Conditions Based on Address Classes on page 39

Firewall Filter Match Conditions Based on Address Fields

You can configure firewall filter match conditions that evaluate packet address fields—IPv4 source and destination addresses, IPv6 source and destination addresses, or media access control (MAC) source and destination addresses—against specified addresses or prefix values.

- Implied Match on the '0/0' Address for Firewall Filter Match Conditions Based on Address Fields on page 33
- Matching an Address Field to a Subnet Mask or Prefix on page 33
- Matching an Address Field to an Excluded Value on page 34
- Matching Either IP Address Field to a Single Value on page 35
- Matching an Address Field to Noncontiguous Prefixes on page 36
- Matching an Address Field to a Prefix List on page 38

Implied Match on the '0/0' Address for Firewall Filter Match Conditions Based on Address Fields

Every firewall filter match condition based on a set of addresses or address prefixes is associated with an implicit match on the address **0.0.0.0/0** (for IPv4 or VPLS traffic) or **0:0:0:0:0:0:0:0/0** (for IPv6 traffic). As a result, any packet whose specified address field does not match any of the specified addresses or address prefixes fails to match the entire term.

Matching an Address Field to a Subnet Mask or Prefix

You can specify a single match condition to match a source address or destination address that falls within a specified address prefix.

IPv4 Subnet Mask Notation

For an IPv4 address, you can specify a subnet mask value rather than a prefix length. For example:

```
[edit firewall family inet filter filter_on_dst_addr term term3 from]
user@host# set address 10.0.0.10/255.0.0.255
```

Prefix Notation

To specify the address prefix, use the notation **prefix/prefix-length**. In the following example, a match occurs if a destination address matches the prefix **10.0.0.0/8**:

```
[edit firewall family inet filter filter_on_dst_addr term term1 from]
user@host# set destination-address 10.0.0.0/8
```

Default Prefix Length for IPv4 Addresses

If you do not specify **/prefix-length** for an IPv4 address, the prefix length defaults to **/32**. The following example illustrates the default prefix value:

```
[edit firewall family inet filter filter_on_dst_addr term term2 from]
user@host# set destination-address 10
user@host# show
destination-address {
    10.0.0.0/32;
}
```

Default Prefix Length for IPv6 Addresses

If you do not specify **/prefix-length** for an IPv6 address, the prefix length defaults to **/128**. The following example illustrates the default prefix value:

```
[edit firewall family inet6 filter filter_on_dst_addr term term1 from]
user@host# set destination-address ::10
user@host# show
destination-address {
    ::10/128;
}
```

Default Prefix Length for MAC Addresses

If you do not specify **/prefix-length** for a media access control (MAC) address of a VPLS, Layer 2 CCC, or Layer 2 bridging packet, the prefix length defaults to **/48**. The following example illustrates the default prefix value:

```
[edit firewall family vpls filter filter_on_dst_mac_addr term term1 from]
```

```
user@host# set destination-mac-address 01:00:0c:cc:cc:cd
user@host# show
destination-address {
    01:00:0c:cc:cc:cd/48;
}
```

Matching an Address Field to an Excluded Value

For the address-field match conditions, you can include the **except** keyword to specify that a match occurs for an address field that does not match the specified address or prefix.

Excluding IP Addresses in IPv4 or IPv6 Traffic

For the following IPv4 and IPv6 match conditions, you can include the **except** keyword to specify that a match occurs for an IP address field that does not match the specified IP address or prefix:

- **address address except**—A match occurs if either the source IP address or the destination IP address does not match the specified address or prefix.
- **source-address address except**—A match occurs if the source IP address does not match the specified address or prefix.
- **destination-address address except**—A match occurs if the destination IP address does not match the specified address or prefix.

In the following example, a match occurs for any IPv4 destination addresses that fall under the **192.168.10.0/8** prefix, except for addresses that fall under **192.168.0.0/16**. All other addresses implicitly do not match this condition.

```
[edit firewall family inet filter filter_on_dst_addr term term1 from]
user@host# set 192.168.0.0/16 except
user@host# set 192.168.10.0/8
user@host# show
destination-address {
    192.168.0.0/16 except;
    192.168.10.0/8;
}
```

In the following example, a match occurs for any IPv4 destination address that does not fall within the prefix **10.1.1.0/24**:

```
[edit firewall family inet filter filter_on_dst_addr term term24 from]
user@host# set destination-address 0.0.0.0/0
user@host# set destination-address 10.1.1.0/24 except
user@host# show
destination-address {
    0.0.0.0/0;
    10.1.1.0/24 except;
}
```

Excluding IP Addresses in VPLS or Layer 2 Bridging Traffic

For the following VPLS and Layer 2 bridging match conditions on MX Series routers only, you can include the **except** keyword to specify that a match occurs for an IP address field that does not match the specified IP address or prefix:

- **ip-address address except**—A match occurs if either the source IP address or the destination IP address does not match the specified address or prefix.
- **source-ip-address address except**—A match occurs if the source IP address does not match the specified address or prefix.
- **destination-ip-address address except**—A match occurs if the destination IP address does not match the specified address or prefix.

In the following example for filtering VPLS traffic on an MX Series router, a match occurs if the source IP address falls within the exception range of **55.0.1.0/255.0.255.0** and the destination IP address matches **55.0.0.0/8**:

```
[edit]
firewall {
  family vpls {
    filter fvpls {
      term 1 {
        from {
          ip-address {
            55.0.0.0/8;
            55.0.1.0/255.0.255.0 except;
          }
        }
        then {
          count from-55/8;
          discard;
        }
      }
    }
  }
}
```

Excluding MAC Addresses in VPLS or Layer 2 Bridging Traffic

For the following VPLS or Layer 2 bridging traffic match conditions, you can include the **except** keyword to specify that a match occurs for a MAC address field that does not match the specified MAC address or prefix:

- **source-mac-address address except**—A match occurs if the source MAC address does not match the specified address or prefix.
- **destination-mac-address address except**—A match occurs if either the destination MAC address does not match the specified address or prefix.

Matching Either IP Address Field to a Single Value

For IPv4 and IPv6 traffic and for VPLS and Layer 2 bridging traffic on MX Series routers only, you can use a single match condition to match a single address or prefix value to either the source or destination IP address field.

Matching Either IP Address Field in IPv4 or IPv6 Traffic

For IPv4 or IPv6 traffic, you can use a single match condition to specify the same address or prefix value as the match for either the source or destination IP address field. Instead of creating separate filter terms that specify the same address for the **source-address** and **destination-address** match conditions, you use only the **address** match condition. A match occurs if *either* the source IP address *or* the destination IP address matches the specified address or prefix.

If you use the **except** keyword with the **address** match condition, a match occurs if *both* the source IP address and the destination IP address match the specified value *before* the exception applies.

In a firewall filter term that specifies either the **source-address** or the **destination-address** match condition, you cannot also specify the **address** match condition.

Matching Either IP Address Field in VPLS or Layer 2 Bridging Traffic

For VPLS or Layer 2 bridging traffic on MX Series routers only, you can use a single match condition to specify the same address or prefix value as the match for either the source or destination IP address field. Instead of creating separate filter terms that specify the same address for the **source-ip-address** and **destination-ip-address** match conditions, you use only the **ip-address** match condition. A match occurs if *either* the source IP address *or* the destination IP address matches the specified address or prefix.

If you use the **except** keyword with the **ip-address** match condition, a match occurs if *both* the source IP address and the destination IP address match the specified value *before* the exception applies.

In a firewall filter term that specifies either the **source-ip-address** or the **destination-ip-address** match condition, you cannot also specify the **ip-address** match condition.

Matching an Address Field to Noncontiguous Prefixes

For IPv4 traffic only, specify a single match condition to match the IP source or destination address field to any prefix specified. The prefixes do not need to be contiguous. That is, the prefixes under the **source-address** or **destination-address** match condition do not need to be adjacent or neighboring to one another.

In the following example, a match occurs if a destination address matches either the 10.0.0.0/8 prefix or the 192.168.0.0/32 prefix:

```
[edit firewall family inet filter filter_on_dst_addr term term5 from]
user@host# set destination-address 10.0.0.0/8
user@host# set destination-address 192.168.0.0/32
user@host# show
destination-address {
  destination-address 10.0.0.0/8;
  destination-address 192.168.0.0/32;
}
```

The order in which you specify the prefixes within the match condition is not significant. Packets are evaluated against all the prefixes in the match condition to determine whether a match occurs. If prefixes overlap, longest-match rules are used to determine whether

a match occurs. A match condition of noncontiguous prefixes includes an implicit **0/0 except** statement, which means that any prefix that does not match any prefix included in the match condition is explicitly considered not to match.

Because the prefixes are order-independent and use longest-match rules, longer prefixes subsume shorter ones as long as they are the same type (whether you specify **except** or not). This is because anything that would match the longer prefix would also match the shorter one.

Consider the following example:

```
[edit firewall family inet filter filter_on_src_addr term term1 from]
source-address {
  172.16.0.0/10;
  172.16.2.0/24 except;
  192.168.1.0;
  192.168.1.192/26 except;
  192.168.1.254;
  172.16.3.0/24; # ignored
  10.2.2.2 except; # ignored
}
```

Within the **source-address** match condition, two addresses are ignored. The **172.16.3.0/16** value is ignored because it falls under the address **172.16.0.0/10**, which is the same type. The **10.2.2.2 except** value is ignored because it is subsumed by the implicit **0.0.0.0/0 except** match value.

Suppose the following source IP address are evaluated by this firewall filter:

- Source IP address **172.16.1.2**—This address matches the **172.16.0.0/10** prefix, and thus the action in the **then** statement is taken.
- Source IP address **172.16.2.2**—This address matches the **172.16.2.0/24** prefix. Because this prefix is negated (that is, includes the **except** keyword), an explicit *mismatch* occurs. The next term in the filter is evaluated, if there is one. If there are no more terms, the packet is discarded.
- Source IP address **10.1.2.3**—This address does not match any of the prefixes included in the **source-address** condition. Instead, it matches the implicit **0.0.0.0/0 except** at the end of the list of prefixes configured under the **source-address** match condition, and is considered to be a mismatch.

The **172.16.3.0/24** statement is ignored because it falls under the address **172.16.0.0/10**—both are the same type.

The **10.2.2.2 except** statement is ignored because it is subsumed by the implicit **0.0.0.0/0 except** statement at the end of the list of prefixes configured under the **source-address** match condition.



BEST PRACTICE: When a firewall filter term includes the **from address address** match condition and a subsequent term includes the **from source-address address** match condition for the same address, packets might be processed by the latter term before they are evaluated by any intervening terms. As a

result, packets that should be rejected by the intervening terms might be accepted instead, or packets that should be accepted might be rejected instead.

To prevent this from occurring, we recommend that you do the following. For every firewall filter term that contains the **from address** *address* match condition, replace that term with two separate terms: one that contains the **from source-address** *address* match condition, and another that contains the **from destination-address** *address* match condition.

Matching an Address Field to a Prefix List

You can define a list of IPv4 or IPv6 address prefixes for use in a routing policy statement or in a stateless firewall filter match condition that evaluates packet address fields.

To define a list of IPv4 or IPv6 address prefixes, include the **prefix-list** *prefix-list* statement.

```
prefix-list name {  
  ip-addresses;  
  apply-path path;  
}
```

You can include the statement at the following hierarchy levels:

- [edit policy-options]
- [edit logical-systems *logical-system-name* policy-options]

After you have defined a prefix list, you can use it when specifying a firewall filter match condition based on an IPv4 or IPv6 address prefix.

```
[edit firewall family family-name filter filter-name term term-name]  
from {  
  source-prefix-list {  
    prefix-lists;  
  }  
  destination-prefix-list {  
    prefix-lists;  
  }  
}
```

For an example, see “Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List” on page 100.

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 17
- Firewall Filter Match Conditions Based on Numbers or Text Aliases on page 27
- Firewall Filter Match Conditions Based on Bit-Field Values on page 28
- Firewall Filter Match Conditions Based on Address Classes on page 39

Firewall Filter Match Conditions Based on Address Classes

For IPv4 and IPv6 traffic only, you can use class-based firewall filter conditions to match packet fields based on source class or destination class.

- Source-Class Usage on page 39
- Destination-Class Usage on page 39
- Guidelines for Applying SCU or DCU Firewall Filters to Output Interfaces on page 39

Source-Class Usage

A *source class* is a set of source prefixes grouped together and given a class name. To configure a firewall filter term that matches an IP source address field to one or more source classes, use the **source-class *class-name*** match condition under the **[edit firewall family (inet | inet6) filter *filter-name* term *term-name* from]** hierarchy level.

Source-class usage (SCU) enables you to monitor the amount of traffic originating from a specific prefix. With this feature, usage can be tracked and customers can be billed for the traffic they receive.

Destination-Class Usage

A *destination class* is a set of destination prefixes grouped together and given a class name. To configure a firewall filter term that matches an IP destination address field to one or more destination classes, use the **destination-class *class-name*** match condition at the **[edit firewall family (inet | inet6) filter *filter-name* term *term-name* from]** hierarchy level.

Destination-class usage (DCU) enables you can track how much traffic is sent to a specific prefix in the core of the network originating from one of the specified interfaces.

Note, however, that DCU limits your ability to keep track of traffic moving in the reverse direction. It can account for all traffic that arrives on a core interface and heads toward a specific customer, but it cannot count traffic that arrives on a core interface from a specific prefix.

Guidelines for Applying SCU or DCU Firewall Filters to Output Interfaces

When applying a SCU or DCU firewall filter to an interface, keep the following guidelines in mind:

- Output interfaces—Class-based firewall filter match conditions work only for firewall filters that you apply to output interfaces. This is because the SCU and DCU are determined after route lookup occurs.
- Input interfaces—Although you can specify a source class and destination class for an input firewall filter, the counters are incremented only if the firewall filter is applied on the output interface.
- Output interfaces for tunnel traffic—SCU and DCU are not supported on the interfaces you configure as the output interface for tunnel traffic for transit packets exiting the router through the tunnel.

**Related
Documentation**

- Guidelines for Configuring Standard Firewall Filters on page 17
- Standard Firewall Filter Match Conditions for IPv4 Traffic on page 236
- Standard Firewall Filter Match Conditions for IPv6 Traffic on page 245
- [*Junos OS Source Class Usage Feature Guide*](#)
- Firewall Filter Match Conditions Based on Numbers or Text Aliases on page 27
- Firewall Filter Match Conditions Based on Bit-Field Values on page 28
- Firewall Filter Match Conditions Based on Address Fields on page 32

CHAPTER 4

Introduction to Standard Firewall Filters for Fragment Handling

- Firewall Filters That Handle Fragmented Packets Overview on page 41

Firewall Filters That Handle Fragmented Packets Overview

You can create stateless firewall filters that handle fragmented packets destined for the Routing Engine. By applying these policies to the Routing Engine, you protect against the use of IP fragmentation as a means to disguise TCP packets from a firewall filter.

For example, consider an IP packet that is fragmented into the smallest allowable fragment size of 8 bytes (a 20-byte IP header plus an 8-byte payload). If this IP packet carries a TCP packet, the first fragment (fragment offset of 0) that arrives at the device contains only the TCP source and destination ports (first 4 bytes), and the sequence number (next 4 bytes). The TCP flags, which are contained in the next 8 bytes of the TCP header, arrive in the second fragment (fragment offset of 1).

See RFC 1858, *Security Considerations for IP Fragment Filtering*.

Related Documentation

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Stateless Firewall Filter to Handle Fragments on page 161

CHAPTER 5

Introduction to Standard Firewall Configuration

- Stateless Firewall Filters That Reference Policers Overview on page 43
- Multiple Standard Firewall Filters Applied as a List Overview on page 44
- Guidelines for Applying Multiple Standard Firewall Filters as a List on page 47
- Multiple Standard Firewall Filters in a Nested Configuration Overview on page 48
- Guidelines for Nesting References to Multiple Standard Firewall Filters on page 49
- Interface-Specific Firewall Filter Instances Overview on page 51
- Filtering Packets Received on a Set of Interface Groups Overview on page 52
- Filtering Packets Received on an Interface Set Overview on page 53
- Filter-Based Forwarding Overview on page 53
- Accounting for Standard Firewall Filters Overview on page 55
- System Logging Overview on page 55
- System Logging of Events Generated for the Firewall Facility on page 56
- Logging of Packet Headers Evaluated by a Firewall Filter Term on page 58

Stateless Firewall Filters That Reference Policers Overview

Policing, or rate limiting, is an important component of firewall filters that lets you limit the amount of traffic that passes into or out of an interface.

A stateless firewall filter that references a policer can provide protection from denial-of-service (DOS) attacks. Traffic that exceeds the rate limits configured for the policer is either discarded or marked as lower priority than traffic that conforms to the configured rate limits. Packets can be marked for a lower priority by being set to a specific output queue, set to a specific packet loss priority (PLP) level, or both. When necessary, low-priority traffic can be discarded to prevent congestion.

A policer specifies two types of rate limits on traffic:

- Bandwidth limit—The average traffic rate permitted, specified as a number of bits per second.

- Maximum burst size—The packet size permitted for bursts of data that exceed the bandwidth limit.

Policing uses an algorithm to enforce a limit on average bandwidth while allowing bursts up to a specified maximum value. You can use policing to define specific classes of traffic on an interface and apply a set of rate limits to each class. After you name and configure a policer, it is stored as a template. You can then apply the policer in an interface configuration or, to rate-limit packet-filtered traffic only, in a firewall filter configuration.

For an IPv4 firewall filter term only, you can also specify a *prefix-specific action* as a nonterminating action that applies a policer to the matched packets. A prefix-specific action applies additional matching criteria on the filter-matched packets based on specified address prefix bits and then associates the matched packets with a counter and policer instance for that filter term or for all terms in the firewall filter.

To apply a policer or a prefix action to packet-filtered traffic, you can use the following firewall filter nonterminating actions:

- **policer** *policer-name*
- **three-color-policer (single-rate | two-rate)** *policer-name*
- **prefix-action** *action-name*

Related Documentation

- Standard Firewall Filter Nonterminating Actions on page 270
- Traffic Policing Overview
- Prefix-Specific Counting and Policing Overview

Multiple Standard Firewall Filters Applied as a List Overview

This topic covers the following information:

- The Challenge: Simplify Large-Scale Firewall Filter Administration on page 44
- A Solution: Apply Lists of Firewall Filters on page 45
- Configuration of Multiple Filters for Filter Lists on page 45
- Application of Filter Lists to a Router Interface on page 45
- Interface-Specific Names for Filter Lists on page 46
- How Filter Lists Evaluate Packets on page 46

The Challenge: Simplify Large-Scale Firewall Filter Administration

Typically, you apply a single stateless firewall filter to an interface in the input or output direction or both. However, this approach might not be practical, when you have a device configured with many interfaces. In large environments, you want the flexibility of being able to modify filtering terms common to multiple interfaces without having to reconfigure the filter of every affected interface.

In general, the solution is to apply an effectively “chained” structure of multiple stateless firewall filters to a single interface. You partition your filtering terms into multiple firewall

filters that each perform a filtering task. You can then choose which filtering tasks you want to perform for a given interface and apply the filtering tasks to that interface. In this way, you only manage the configuration for a filtering task in a single firewall filter.

The Junos OS policy framework provides two options for managing the application of multiple separate firewall filters to individual router interfaces. One option is to apply multiple filters as a single input list or output list. The other option is to reference a stateless firewall filter from within the term of another stateless firewall filter.

A Solution: Apply Lists of Firewall Filters

The most straightforward way to avoid configuring duplicate filtering terms common to multiple stateless firewall filters is to configure multiple firewall filters and then apply a customized *list* of filters to each interface. The Junos OS uses the filters—in the order in which they appear in the list—to evaluate packets that transit the interface. If you need to modify filtering terms shared across multiple interfaces, you only need to modify one firewall filter that contains those terms.



NOTE: In contrast with the alternative approach (configuring nested firewall filters) applying firewall filter lists combines multiple firewall filters at each interface application point.

Configuration of Multiple Filters for Filter Lists

Configuring firewall filters to be applied in unique lists for each router interface involves separating shared packet-filtering rules from interface-specific packet-filtering rules as follows:

- **Unique filters**—For each set of packet-filtering rules unique to a specific interface, configure a separate firewall filter that contains only the filtering terms for that interface.
- **Shared filters**—For each set of packet-filtering rules common across two or more interfaces, consider configuring a separate firewall filter that contains the shared filtering terms.



TIP: When planning for a large number firewall filters to be applied using filter lists, administrators often organize the shared filters by filtering criteria, by the services to which customers subscribe, or by the purposes of the interfaces.

Application of Filter Lists to a Router Interface

Applying a list of firewall filters to an interface is a matter of selecting the filters that meet the packet-filtering requirements of that interface. For each interface, you can include an **input-list** or **output-list** statement (or both) within the **filter** stanza to specify the relevant filters in the order in which they are to be used:

- Include any filters that contain common filtering terms relevant to the interface.
- Include the filter that contain only the filtering terms unique to the interface.

Interface-Specific Names for Filter Lists

Because a filter list is configured under an interface, the resulting concatenated filter is *interface-specific*. The system-generated name of an interface-specific filter consists of the full interface name followed by either **'-i'** for an input filter list or **'-o'** for an output filter list.

- **Input filter list name**—For example, if you use the **input-list** statement to apply a chain of filters to logical interface **ge-1/3/0.0**, the Junos OS uses the following name for the filter:

ge-1/3/0.0-i

- **Output filter list name**—For example, if you use the **output-list** statement to apply a chain of filters to logical interface **fe-0/1/2.0**, the Junos OS uses the following name for the filter:

fe-0/1/2.0-o

You can use the interface-specific name of a filter list when you enter a Junos OS operational mode command that specifies a stateless firewall filter name.

How Filter Lists Evaluate Packets

The policy framework software evaluates a packet against the filters in a list sequentially, beginning with the first filter in the list until either a terminating action occurs or the packet is implicitly discarded:

- If the packet matches a filter term that specifies a terminating action, any subsequent filters in the list are not used to evaluate the packet. Terminating actions include the following: **accept**, **discard**, **reject**, **logical-system *logical-system-name***, **routing-instance *routing-instance-name***, and **topology *topology-name***.
- If the packet matches a filter term that does not specify either a terminating action or the **next term** action, any nonterminating actions are taken and then the packet is implicitly accepted. The **accept** action is a terminating action, and therefore any subsequent filters in the list are not used to evaluate the packet.

If the packet matches a term that includes the **next term** action, the matching packet is evaluated against the next term in the firewall filter; otherwise, the matching packet is not evaluated against subsequent terms in the firewall filter. For example, if a packet matches a standard firewall filter term that is configured with the **count** nonterminating action and the **next term** flow control action, the packet is counted and evaluation of the packet continues with the next term or firewall filter.

- Otherwise, if no packet match results in an implicit or explicit terminating action, the packet is implicitly discarded.

Related Documentation

- How Standard Firewall Filters Evaluate Packets on page 16
- Guidelines for Applying Multiple Standard Firewall Filters as a List on page 47
- Example: Applying Lists of Multiple Standard Firewall Filters on page 171

Guidelines for Applying Multiple Standard Firewall Filters as a List

This topic covers the following information:

- Statement Hierarchy for Applying Lists of Multiple Firewall Filters on page 47
- Filter Input Lists and Output Lists for Router Interfaces on page 47
- Types of Filters Supported in Lists on page 47
- Restrictions on Applying Filter Lists for MPLS or Layer 2 CCC Traffic on page 48

Statement Hierarchy for Applying Lists of Multiple Firewall Filters

To apply a single filter to the input or output direction of a router logical interface, you include the **input** *filter-name* or **output** *filter-name* statement under the **filter** stanza for a protocol family.

To apply a list of multiple filters to the input or output direction of a router logical interface, include the **input-list** [*filter-names*] or **output-list** [*filter-names*] statement under the **filter** stanza for a protocol family:

```

interfaces {
  interface-name {
    unit logical-unit-number {
      family family-name {
        filter {
          ...filter-options...
          input-list [ filter-names ];
          output-list [ filter-names ];
        }
      }
    }
  }
}

```

You can include the interface configuration at one of the following hierarchy levels:

- [edit]
- [edit logical-systems *logical-system-name*]

Filter Input Lists and Output Lists for Router Interfaces

When applying a list of firewall filters as a list, the following limitations apply:

- You can specify up to 16 firewall filters for a filter input list.
- You can specify up to 16 firewall filters for a filter output list.

Types of Filters Supported in Lists

Lists of multiple firewall filters applied to a router interface support standard stateless firewall filters only. You cannot apply lists containing service filters or simple filters to a router interface.

Restrictions on Applying Filter Lists for MPLS or Layer 2 CCC Traffic

When applying stateless firewall filters that evaluate MPLS traffic (**family mpls**) or Layer 2 circuit cross-connection traffic (**family ccc**), you can use the **input-list [filter-names]** and **output-list [filter-names]** statements for all interfaces except the following:

- Management and internal Ethernet (**fxp**) interfaces
- Loopback (**lo0**) interfaces
- USB modem (**umd**) interfaces

**Related
Documentation**

- Multiple Standard Firewall Filters Applied as a List Overview on page 44
- Example: Applying Lists of Multiple Standard Firewall Filters on page 171

Multiple Standard Firewall Filters in a Nested Configuration Overview

This topic covers the following information:

- The Challenge: Simplify Large-Scale Firewall Filter Administration on page 48
- A Solution: Configure Nested References to Firewall Filters on page 48
- Configuration of Nested Firewall Filters on page 49
- Application of Nested Firewall Filters to a Router Interface on page 49

The Challenge: Simplify Large-Scale Firewall Filter Administration

Typically, you apply a single stateless firewall filter to an interface in the input or output direction or both. This approach might not be practical, however, when you have a router configured with many, even hundreds of interfaces. In an environment of this scale, you want the flexibility of being able to modify filtering terms common to multiple interfaces without having to reconfigure the filter of every affected interface.

In general, the solution is to apply an effectively “chained” structure of multiple stateless firewall filters to a single interface. You partition your filtering terms into multiple firewall filters configured so that you can apply a unique filter to each router interface but also apply common filters to multiple router interfaces as required. The Junos OS policy framework provides two options for managing the application of multiple separate firewall filters to individual router interfaces. One option is to apply multiple filters as a single input list or output list. The other option is to reference a stateless firewall filter from within the term of another stateless firewall filter.

A Solution: Configure Nested References to Firewall Filters

The most structured way to avoid configuring duplicate filtering terms common to multiple stateless firewall filters is to configure multiple stateless firewall filters so that each filter includes the shared filtering terms by *referencing* a separate filter that contains the common filtering terms. The Junos OS uses the filter terms—in the order in which they appear in the filter definition—to evaluate packets that transit the interface. If you need to modify filtering terms shared across multiple interfaces, you only need to modify one firewall filter.



NOTE: Similar to the alternative approach (applying a list of firewall filters), configuring a nested firewall filter combines multiple firewall filters into a new firewall filter definition.

Configuration of Nested Firewall Filters

Configuring a nested firewall filter for each router interface involves separating shared packet-filtering rules from interface-specific packet-filtering rules as follows:

- For each set of packet-filtering rules common across multiple interfaces, configure a separate firewall filter that contains the shared filtering terms.
- For each router interface, configure a separate firewall filter that contains:
 - All the filtering terms unique to that interface.
 - An additional filtering term that includes a **filter** reference to the firewall filter that contains the common filtering terms.

Application of Nested Firewall Filters to a Router Interface

Applying nested firewall filters is no different from applying an unnested firewall filter. For each interface, you can include an **input** or **output** statement (or both) within the **filter** stanza to specify the appropriate nested firewall filter.

Applying nested firewall filters to an interface, the shared filtering terms and the interface-specific firewall filters are applied through a *single nested firewall filter* that includes other filters through the **filter** statement within a separate filtering term.

Related Documentation

- Guidelines for Nesting References to Multiple Standard Firewall Filters on page 49
- Example: Nesting References to Multiple Standard Firewall Filters on page 176

Guidelines for Nesting References to Multiple Standard Firewall Filters

This topic covers the following information:

- Statement Hierarchy for Configuring Nested Firewall Filters on page 49
- Filter-Defining Terms and Filter-Referencing Terms on page 50
- Types of Filters Supported in Nested Configurations on page 50
- Number of Filter References in a Single Filter on page 50
- Depth of Filter Nesting on page 50

Statement Hierarchy for Configuring Nested Firewall Filters

To reference a filter from within a filter, include the **filter filter-name** statement as a separate filter term:

```
firewall firewall-name {
  family family-name {
    filter filter-name {
      term term-name {
```

```
        filter filter-name;
      }
    }
  }
}
```

You can include the firewall configuration at one of the following hierarchy levels:

- `[edit]`
- `[edit logical-systems logical-system-name]`

Filter-Defining Terms and Filter-Referencing Terms

You cannot configure a firewall filter term that both references another firewall filter and defines a match condition or action. If a firewall filter term includes the **filter** statement, then it cannot also include the **from** or **then** statement.

For example, the firewall filter term **term term1** in the configuration is *not* valid:

```
[edit]
firewall {
  family inet {
    filter filter_1 {
      term term1 {
        filter filter_2;
        from {
          source-address 1.1.1.1/32;
        }
        then {
          accept;
        }
      }
    }
  }
}
```

In order for **term term1** to be a valid filter term, you must either remove the **filter filter_2** statement or remove both the **from** and **then** stanzas.

Types of Filters Supported in Nested Configurations

Nested configurations of firewall filters support standard stateless firewall filters only. You cannot use service filters or simple filters in a nested firewall filter configuration.

Number of Filter References in a Single Filter

The total number of filters referenced from within a filter cannot exceed 256.

Depth of Filter Nesting

The Junos OS supports a single level of firewall filter nesting. If **filter_1** references **filter_2**, you cannot configure a filter that references a filter that references **filter_1**.

Related Documentation

- Multiple Standard Firewall Filters in a Nested Configuration Overview on page 48
- Example: Nesting References to Multiple Standard Firewall Filters on page 176

Interface-Specific Firewall Filter Instances Overview

This topic covers the following information:

- Instantiation of Interface-Specific Firewall Filters on page 51
- Interface-Specific Names for Firewall Filter Instances on page 51
- Interface-Specific Firewall Filter Counters on page 52
- Interface-Specific Firewall Filter Policers on page 52

Instantiation of Interface-Specific Firewall Filters

On T Series, M120, M320, and MX Series routers, you can enable the Junos OS to automatically create an interface-specific instance of a firewall filter for each interface to which you apply the filter. If you enable interface-specific instantiation of a firewall filter and then apply that filter to multiple interfaces, any **count** actions or **policer** actions configured in the filter terms act on the traffic stream entering or exiting each individual interface, regardless of the sum of traffic on the multiple interfaces.

You can enable this option per firewall filter by including the **interface-specific** statement in the filter configuration.



NOTE: On T Series, M120, M320, and MX Series routers, interfaces are distributed among multiple packet-forwarding components.

Interface-specific firewall filtering is not supported on M Series routers other than the M120 and M320 routers. If you apply a firewall filter to multiple interfaces on an M Series router other than the M120 or M320 routers, the filter acts on the sum of traffic entering or exiting those interfaces.

Interface-specific firewall filtering is supported for standard stateless firewall filters and for service filters. Interface-specific instances are not supported for simple filters.

Interface-Specific Names for Firewall Filter Instances

When the Junos OS creates a separate instance of a firewall filter for a logical interface, the instance is associated with an interface-specific name. The system-generated name of a firewall filter instance consists of the name of the configured filter followed by a hyphen ('-'), the full interface name, and either '-i' for an input filter instance or '-o' for an output filter instance.

- **Input filter instance name**—For example, if you apply the interface-specific firewall filter **filter_s_tcp** to the input at logical interface **at-1/1/1.0**, the Junos OS instantiates an interface-specific filter instance with the following system-generated name:

filter_s_tcp-at-1/1/1.0-i

- **Output filter instance name**—For example, if you apply the interface-specific firewall filter **filter_s_tcp** to the output at logical interface **so-2/2/2.2**, the Junos OS instantiates an interface-specific filter instance with the following system-generated name:

count_s_tcp-so-2/2/2.2-o

You can use the interface-specific name of a filter instance when you enter a Junos OS operational mode command that specifies a stateless firewall filter name.



TIP: When you configure a firewall filter with interface-specific instances enabled, we recommend you limit the filter name to *52 bytes* in length. This is because firewall filter names are restricted to *64 bytes* in length. If a system-generated filter instance name exceeds this maximum length, the policy framework software might reject the instance name.

Interface-Specific Firewall Filter Counters

Instantiation of interface-specific firewall filters causes the Packet Forwarding Engine to maintain any counters for the firewall filter separately for each interface. You specify interface-specific counters per firewall filter term by specifying the **count counter-name** non-terminating action.

The system-generated name of an interface-specific firewall filter counter consists of the name of the configured counter followed by a hyphen ('-'), the full interface name, and either '-i' for an input filter instance or '-o' for an output filter instance.

- **Interface-specific input filter counter name**—For example, suppose you configure the filter counter **count_tcp** for an interface-specific firewall filter. If the filter is applied to the input at logical interface **at-1/1/1.0**, the Junos OS creates the following system-generated counter name:

count_tcp-at-1/1/1.0-i

- **Interface-specific output filter counter name**—For example, suppose you configure the filter counter **count_udp** for an interface-specific firewall filter. If the filter is applied to the output at logical interface **so-2/2/2.2**, the Junos OS creates the following system-generated counter name:

count_udp-so-2/2/2.2-o

Interface-Specific Firewall Filter Policers

Instantiation of interface-specific firewall filters not only creates separate instances of any firewall filter counters but also creates separate instances of any policer actions. Any policers applied through an action specified in the firewall filter configuration are applied separately to each interface in the interface group. You specify interface-specific policers per firewall filter term by specifying the **policer policer-name** non-terminating action.

Related Documentation

- Statement Hierarchy for Configuring Interface-Specific Firewall Filters on page 289
- Statement Hierarchy for Applying Interface-Specific Firewall Filters on page 290
- Example: Configuring Interface-Specific Firewall Filter Counters on page 180

Filtering Packets Received on a Set of Interface Groups Overview

You can configure a firewall filter term that matches packets tagged for a specified *interface group* or set of interface groups. An interface group consists of one or more

logical interfaces with the same group number. Packets received on an interface in an interface group are tagged as being part of that group.

For standard stateless firewall filters, you can filter packets received on an interface group for IPv4, IPv6, virtual private LAN service (VPLS), Layer 2 circuit cross-connection (CCC), and Layer 2 bridging traffic. For service filters, you can filter packets received on an interface group for either IPv4 or IPv6 traffic.



NOTE: You can also configure a firewall filter term that matches on packets tagged for a specified *interface set*. For more information, see “Filtering Packets Received on an Interface Set Overview” on page 53.

**Related
Documentation**

- Statement Hierarchy for Assigning Interfaces to Interface Groups on page 291
- Statement Hierarchy for Configuring a Filter to Match on a Set of Interface Groups on page 291
- Example: Filtering Packets Received on an Interface Group on page 184

Filtering Packets Received on an Interface Set Overview

You can configure a standard stateless firewall filter term that matches packets tagged for a specified *interface set*. An interface set groups two or more physical or logical interfaces into a single interface-set name. You can filter packets received on an interface set for protocol-independent, IPv4, IPv6, MPLS, VPLS, or bridging traffic.



NOTE: You can also configure a standard stateless firewall filter term or a service filter term that matches on packets tagged for a specified *interface group*. For more information, see “Filtering Packets Received on a Set of Interface Groups Overview” on page 52.

**Related
Documentation**

- Statement Hierarchy for Defining an Interface Set on page 293
- Statement Hierarchy for Configuring a Filter to Match on an Interface Set on page 293
- Example: Configuring a Rate-Limiting Filter Based on Destination Class on page 167
- Example: Filtering Packets Received on an Interface Set on page 188

Filter-Based Forwarding Overview

- Filters That Classify Packets or Direct Them to Routing Instances on page 54
- Input Filtering to Classify and Forward Packets Within the Router on page 54
- Output Filtering to Forward Packets to Another Routing Table on page 54
- Restrictions for Applying Filter-Based Forwarding on page 54

Filters That Classify Packets or Direct Them to Routing Instances

For IPv4, IPv6, or MPLS-tagged IPv4 traffic only, you can use stateless firewall filters in conjunction with forwarding classes and routing instances to control how packets travel in a network. This is called *filter-based forwarding* (FBF).

You can define a filtering term that matches incoming packets based on source address and then classifies matching packets to a specified forwarding class. This type of filtering can be configured to grant certain types of traffic preferential treatment or to improve load balancing. To configure a stateless firewall filter to classify packets to a forwarding class, configure a term with the *nonterminating action forwarding-class class-name*.

You can also define a filtering term that directs matching packets to a specified routing instance. This type of filtering can be configured to route specific types of traffic through a firewall or other security device before the traffic continues on its path. To configure a stateless firewall filter to direct traffic to a routing instance, configure a term with the *terminating action routing-instance routing-instance-name <topology topology-name>* to specify the routing instance to which matching packets will be forwarded.

Input Filtering to Classify and Forward Packets Within the Router

You can configure filters to classify packets based on source address and specify the forwarding path the packets take within the router by configuring a filter on the ingress interface.

For example, you can use this filter for applications to differentiate traffic from two clients that have a common access layer (for example, a Layer 2 switch) but are connected to different Internet service providers (ISPs). When the filter is applied, the router can differentiate the two traffic streams and direct each to the appropriate network. Depending on the media type the client is using, the filter can use the source IP address to forward the traffic to the corresponding network through a tunnel. You can also configure filters to classify packets based on IP protocol type or IP precedence bits.

Output Filtering to Forward Packets to Another Routing Table

You can also forward packets based on output filters by configuring a filter on the egress interfaces. In the case of port mirroring, it is useful for port-mirrored packets to be distributed to multiple monitoring PICs and collection PICs based on patterns in packet headers. FBF on the port-mirroring egress interface must be configured.

Packets forwarded to the output filter have been through at least one route lookup when an FBF filter is configured on the egress interface. After the packet is classified at the egress interface by the FBF filter, it is redirected to another routing table for further route lookup.

Restrictions for Applying Filter-Based Forwarding

An interface configured with filter-based forwarding does not support source-class usage (SCU) filter matching and unicast reverse-path forwarding (RPF) check filters.

Related Documentation

- Statement Hierarchy for Configuring FBF for IPv4 or IPv6 Traffic on page 294
- Statement Hierarchy for Configuring FBF for MPLS-Tagged IPv4 Traffic on page 295
- Statement Hierarchy for Configuring Routing Instances for FBF on page 297

- Statement Hierarchy for Applying FBF Filters to Interfaces on page 298
- Example: Configuring Filter-Based Forwarding on the Source Address on page 194

Accounting for Standard Firewall Filters Overview

Juniper Networks devices can collect various kinds of data about traffic passing through the device. You can set up one or more accounting profiles that specify some common characteristics of this data, including the following:

- Fields used in the accounting records.
- Number of files that the routing platform retains before discarding, and the number of bytes per file.
- Polling period that the system uses to record the data

There are several types of accounting profiles: interface, firewall filter, source class and destination class usage, and Routing Engine. If you apply the same profile name to both a firewall filter and an interface, it causes an error.

Related Documentation

- Statement Hierarchy for Configuring Firewall Filter Accounting Profiles on page 299
- Statement Hierarchy for Applying Firewall Filter Accounting Profiles on page 300
- Example: Configuring Statistics Collection for a Standard Firewall Filter on page 201

System Logging Overview

The Junos OS generates system log messages (also called *syslog messages*) to record *system events* that occur on the device. Events consist of routine operations, failure and error conditions, and critical conditions that might require urgent resolution. This system logging utility is similar to the UNIX **syslogd** utility.

Each Junos OS system log message belongs to a message category, called a *facility*, that reflects the hardware- or software-based source of the triggering event. A group of messages belonging to the same facility are either generated by the same software process or concern a similar hardware condition or user activity (such as authentication attempts). Each system log message is also preassigned a *severity*, which indicates how seriously the triggering event affects router functions. Together, the facility and severity of an event are known as the message *priority*. The content of a syslog message identifies the Junos OS *process* that generates the message and briefly describes the operation or error that occurred.

By default, syslog messages that have a severity of **info** or more serious are written to the main system log file **messages** in the **/var/log** directory of the local Routing Engine. To configure global settings and facility-specific settings that override these default values, you can include statements at the **[edit system syslog]** hierarchy level.

For all syslog facilities or for a specified facility, you can configure the syslog message utility to redirect messages of a specified severity to a specified file instead of to the main

system log file. You can also configure the syslog message utility to write syslog messages of a specified severity, for all syslog facilities or for a specified facility, to additional destinations. In addition to writing syslog messages to a log file, you can write syslog messages to the terminal sessions of any logged-in users, to the router console, or to a remote host or the other Routing Engine.

At the global level—for all system logging messages, regardless of facility, severity, or destination—you can override the default values for file-archiving properties and the default timestamp format.

**Related
Documentation**

- System Logging of Events Generated for the Firewall Facility on page 56
- Logging of Packet Headers Evaluated by a Firewall Filter Term on page 58
- Example: Configuring Logging for a Stateless Firewall Filter Term on page 206

System Logging of Events Generated for the Firewall Facility

System log messages generated for firewall filter actions belong to the **firewall** facility. Just as you can for any other Junos OS system logging facility, you can direct **firewall** facility syslog messages to one or more specific destinations: to a specified file, to the terminal session of one or more logged in users (or to all users), to the router console, or to a remote host or the other Routing Engine on the router.

When you configure a syslog message destination for **firewall** facility syslog messages, you include a statement at the **[edit system syslog]** hierarchy level, and you specify the **firewall** facility name together with a severity level. Messages from the **firewall** that are rated at the specified level or more severe are logged to the destination.

System log messages with the **DFWD_** prefix are generated by the firewall process (**dfwd**), which manages compilation and downloading of Junos OS firewall filters. System log messages with the **PFE_FW_** prefix are messages about firewall filters, generated by the Packet Forwarding Engine controller, which manages packet forwarding functions. For more information, see the [Junos OS System Log Messages Reference](#).

Table 9 on page 57 lists the system log destinations you can configure for the **firewall** facility.

Table 9: Syslog Message Destinations for the Firewall Facility

Destination	Description	Configuration Statements
File	<p>Configuring this option keeps the firewall syslog messages out of the main system log file.</p> <p>To include priority and facility with messages written to the file, include the explicit-priority statement.</p> <p>To override the default standard message format, which is based on a UNIX system log format, include the structured-data statement.[†]</p>	<pre>[edit] system { syslog { file <i>filename</i> { firewall severity; allow-duplicates; # File option archive <i>archive-options</i>; # File option explicit-priority; # File option structured-data; # File option } allow-duplicates; # All destinations archive <i>archive-options</i>; # All files time-format (<i>option</i>); # Local destinations } }</pre>
Terminal session	<p>Configuring this option causes a copy of the firewall syslog messages to be written to the specified terminal sessions. Specify one or more user names, or specify * for all logged in users.</p>	<pre>[edit] system { syslog { user (<i>username</i> *) { firewall severity; } time-format (<i>option</i>); # Local destinations } }</pre>
Router console	<p>Configuring this option causes a copy of the firewall syslog messages to be written to the router console.</p>	<pre>[edit] system { syslog { console { firewall severity; } time-format (<i>option</i>); # Local destinations } }</pre>
Remote host or the other Routing Engine	<p>Configuring this option causes a copy of the firewall syslog messages to be written to the specified remote host or to the other Routing Engine.</p> <p>To override the default alternative facility for forwarding firewall syslog messages to a remote machine (local3), include the facility-override firewall statement.</p> <p>To include priority and facility with messages written to the file, include the explicit-priority statement.</p>	<pre>[edit] system { syslog { host (<i>hostname</i> other-routing-engine) { firewall severity; allow-duplicates; # Host option archive <i>archive-options</i>; # File option facility-override firewall; # Host option explicit-priority; # Host option } allow-duplicates; # All destinations archive <i>archive-options</i>; # All files time-format (<i>option</i>); # Local destinations } }</pre>

[†] When the **structured-data** statement is included, other statements that specify the format for messages written to the file are ignored (the **explicit-priority** statement at the [edit system syslog file *filename*] hierarchy level and the **time-format** statement at the [edit system syslog] hierarchy level).

By default, the timestamp recorded in a standard-format system log message specifies the month, date, hour, minute, and second when the message was logged, as in the example:

Sep 07 08:00:10

To include the year, the millisecond, or both in the timestamp for all system logging messages, regardless of the facility, include one of the following statement at the **[edit system syslog]** hierarchy level:

- **time-format year;**
- **time-format millisecond;**
- **time-format year millisecond;**

The following example illustrates the format for a timestamp that includes both the millisecond (401) and the year (2010):

Sep 07 08:00:10.401.2010

Related Documentation

- System Logging Overview on page 55
- Logging of Packet Headers Evaluated by a Firewall Filter Term on page 58
- Example: Configuring Logging for a Stateless Firewall Filter Term on page 206
- “Junos OS System Logging Facilities and Message Severity Levels” in the *Junos OS System Basics Configuration Guide*
- “Junos OS System Log Configuration Statements” in the *Junos OS System Basics Configuration Guide*
- “Junos OS Default System Log Settings” in the *Junos OS System Basics Configuration Guide*
- “Logging Messages in Structured-Data Format” in the *Junos OS System Basics Configuration Guide*
- “Including the Year or Millisecond in Timestamps” in the *Junos OS System Basics Configuration Guide*
- “Changing the Alternative Facility Name for Remote System Log Messages” in the *Junos OS System Basics Configuration Guide*
- “Junos OS System Log Alternate Facilities for Remote Logging” in the *Junos OS System Basics Configuration Guide*

Logging of Packet Headers Evaluated by a Firewall Filter Term

Built in to the stateless firewall filtering software is the capability to log packet-header information for the packets evaluated by a stateless firewall filter term. You can write the packet header information to the system log file on the local Routing Engine or to a firewall filter buffer in the Packet Forwarding Engine. Logging of packet headers evaluated by firewall filters is supported for standard stateless firewall filters for IPv4 or IPv6 traffic only. Service filters and simple filters do not support logging of packet headers.

Table 10 on page 59 lists the packet-header logs you can configure for a firewall filter action.

Table 10: Packet-Header Logs for Stateless Firewall Filter Terms

Log	Description	Configuration Statements
Syslog message destinations configured for the firewall facility	<p>Configure this option by using the syslog nonterminating action.</p> <p>NOTE: Packet header information is interspersed with event messages.</p> <p>To list log files, enter the show log operational mode command without command options.</p> <p>To display log file contents for a specific file in the /var/log directory on the local Routing Engine, enter the show log filename operational mode command or the file show /var/log/filename operational mode command.</p> <p>To clear log file contents, enter the clear log filename <all> operational mode command. If you include the all option, the specified log file is truncated, all archived versions of the log file are deleted.</p>	<pre>firewall { family { filter filter-name { from { match-conditions; } then { ... syslog; terminating-action; } } } }</pre>
Buffer in the Packet Forwarding Engine	<p>Configure this option by using the log nonterminating action.</p> <p>NOTE: Restarting the router causes the contents of this buffer to be cleared.</p> <p>To display the local log entries for firewall filters, enter the show firewall log operational mode command.</p>	<pre>firewall { family { filter filter-name { from { match-conditions; } then { ... log; terminating-action; } } } }</pre>

- Related Documentation**
- System Logging Overview on page 55
 - System Logging of Events Generated for the Firewall Facility on page 56
 - Example: Configuring Logging for a Stateless Firewall Filter Term on page 206

CHAPTER 6

Introduction to Service Filter Configuration

- Service Filter Overview on page 61
- How Service Filters Evaluate Packets on page 62
- Guidelines for Configuring Service Filters on page 64
- Guidelines for Applying Service Filters on page 66

Service Filter Overview

This topic covers the following information:

- Services on page 61
- Service Rules on page 61
- Service Rule Refinement on page 61
- Service Filter Counters on page 62

Services

The Adaptive Services Physical Interface Cards (PICs), Multiservices PICs, and Multiservices Dense Port Concentrators (DPCs) provide *adaptive services interfaces*. Adaptive services interfaces enable you to coordinate a special range of services on a single PIC or DPC by configuring a set of services and applications.



NOTE: Service filters are not supported on J Series devices and Branch SRX devices.

Service Rules

A *service set* is an optional definition you can apply to the traffic at an adaptive services interface. A service set enables you to configure combinations of directional rules and default settings that control the behavior of each service in the service set.

Service Rule Refinement

When you apply a service set to the traffic at an adaptive services interface, you can optionally use *service filters* to refine the target of the set of services and also to process traffic. Service filters enable you to manipulate traffic by performing packet filtering to a defined set of services on an adaptive services interface before the traffic is delivered

to its destination. You can apply a service filter to traffic before packets are accepted for input or output service processing or after packets return from input service processing.

Service Filter Counters

Like standard firewall filters, service filters support counting of matched packets. When you display counters for a service filter, however, the syntax for specifying the filter name includes the name of the *service set* to which the service filter is applied.

- To enable counting of the packets matched by a service filter term, specify the **count *counter-name*** nonterminating action in that term.
- To display counters for service filters, use the **show firewall filter *filter-name* <counter *counter-name*>** operational mode command, and specify the *filter-name* as follows:

__service-service-set-name:service-filter-name

For example, suppose you configure a service filter named **out_filter** with a counter named **out_counter** and apply that service filter to a logical interface to direct certain packets for processing by the output services associated with the service set **nat_set**. In this scenario, the syntax for using the **show firewall** operational mode command to display the counter is as follows:

[edit]

user@host> **show firewall filter *__service-nat_set:out_filter* counter out_counter**

Related Documentation

- Stateless Firewall Filter Types on page 6
- How Service Filters Evaluate Packets on page 62
- Guidelines for Configuring Service Filters on page 64
- Guidelines for Applying Service Filters on page 66
- Example: Configuring and Applying Service Filters on page 211
- “Adaptive Services Overview” in the *Junos OS Services Interfaces Configuration Guide*
- “Configuring Service Sets to be Applied to Services Interfaces” in the *Junos OS Services Interfaces Configuration Guide*
- “Configuring Service Rules” in the *Junos OS Services Interfaces Configuration Guide*

How Service Filters Evaluate Packets

This topic covers the following information:

- Service Filters That Contain a Single Term on page 63
- Service Filters That Contain Multiple Terms on page 63
- Service Filter Terms That Do Not Contain Any Match Conditions on page 63
- Service Filter Terms That Do Not Contain Any Actions on page 63
- Service Filter Default Action on page 63

Service Filters That Contain a Single Term

For a service filter that consists of a single term, the policy framework software evaluates a packet as follows:

- If the packet matches all the conditions, the actions are taken.
- If the packet matches all the conditions and no actions are specified, the packet is accepted.
- If the packet does not match all the conditions, it is discarded.

Service Filters That Contain Multiple Terms

For a service filter that consists of multiple terms, the policy framework software evaluates a packet against the terms in the filter sequentially, beginning with the first term in the filter, until either the packet matches all the conditions in one of the terms or there are no more terms in the filter.

- If the packet matches all the conditions in a term, the actions in that term are performed and evaluation of the packet ends at that term. Any subsequent terms in the filter are not used.
- If the packet does not match all the conditions in the term, evaluation of the packet proceeds to the next term in the filter.

Service Filter Terms That Do Not Contain Any Match Conditions

For service filters with a single term and for filters with multiple terms, if a term does not contain any match conditions, the actions are taken on any packet evaluated.

Service Filter Terms That Do Not Contain Any Actions

If a term does not contain any actions, and if the packet matches the conditions in the term, the packet is accepted.

Service Filter Default Action

Each service filter has an *implicit skip* action at the end of the filter, which is equivalent to including the following example term **explicit_skip** as the final term in the service filter:

```
term explicit_skip {  
    then skip;  
}
```

By default, if a packet matches none of the terms in a service filter, the packet bypasses service processing.

Related Documentation

- Service Filter Overview on page 61
- Guidelines for Configuring Service Filters on page 64
- Guidelines for Applying Service Filters on page 66
- Example: Configuring and Applying Service Filters on page 211

Guidelines for Configuring Service Filters

This topic covers the following information:

- Statement Hierarchy for Configuring Service Filters on page 64
- Service Filter Protocol Families on page 64
- Service Filter Names on page 64
- Service Filter Terms on page 64
- Service Filter Match Conditions on page 65
- Service Filter Terminating Actions on page 65

Statement Hierarchy for Configuring Service Filters

To configure a service filter, include the **service-filter service-filter-name** statement at the **[edit firewall family (inet | inet6)]** hierarchy level:

```
[edit]
firewall {
  family (inet | inet6) {
    service-filter service-filter-name {
      term term-name {
        from {
          match-conditions;
        }
        then {
          actions;
        }
      }
    }
  }
}
```

Individual statements supported under the **service-filter service-filter-name** statement are described separately in this topic and are illustrated in the example of configuring and applying a service filter.

Service Filter Protocol Families

You can configure service filters to filter IPv4 traffic (**family inet**) and IPv6 traffic (**family inet6**) only. No other protocol families are supported for service filters.

Service Filter Names

Under the **family inet** or **family inet6** statement, you can include **service-filter service-filter-name** statements to create and name service filters. The filter name can contain letters, numbers, and hyphens (-) and be up to 64 characters long. To include spaces in the name, enclose the entire name in quotation marks (" ").

Service Filter Terms

Under the **service-filter service-filter-name** statement, you can include **term term-name** statements to create and name filter terms.

- You must configure at least one term in a firewall filter.

- You must specify a unique name for each term within a firewall filter. The term name can contain letters, numbers, and hyphens (-) and can be up to 64 characters long. To include spaces in the name, enclose the entire name in quotation marks (" ").
- The order in which you specify terms within a firewall filter configuration is important. Firewall filter terms are evaluated in the order in which they are configured. By default, new terms are always added to the end of the existing filter. You can use the **insert** configuration mode command to reorder the terms of a firewall filter.

Service Filter Match Conditions

Service filter terms support only a subset of the IPv4 and IPv6 match conditions that are supported for standard stateless firewall filters.

If you specify an IPv6 address in a match condition (the **address**, **destination-address**, or **source-address** match conditions), use the syntax for text representations described in RFC 2373, *IP Version 6 Addressing Architecture*. For more information about IPv6 addresses, see "IPv6 Overview" and "IPv6 Standards" in the [Junos OS Routing Protocols Configuration Guide](#).

Service Filter Terminating Actions

When configuring a service filter term, you must specify one of the following filter-terminating actions:

- **service**
- **skip**



NOTE: These actions are unique to service filters.

Service filter terms support only a subset of the IPv4 and IPv6 nonterminating actions that are supported for standard stateless firewall filters:

- **count** *counter-name*
- **log**
- **port-mirror**
- **sample**

Service filters do not support the **next** action.

Related Documentation

- Service Filter Overview on page 61
- How Service Filters Evaluate Packets on page 62
- Guidelines for Applying Service Filters on page 66
- Service Filter Match Conditions for IPv4 or IPv6 Traffic on page 275
- Service Filter Terminating Actions on page 281
- Service Filter Nonterminating Actions on page 282

- Example: Configuring and Applying Service Filters on page 211

Guidelines for Applying Service Filters

This topic covers the following information:

- Restrictions for Adaptive Services Interfaces on page 66
- Statement Hierarchy for Applying Service Filters on page 66
- Associating Service Rules with Adaptive Services Interfaces on page 67
- Filtering Traffic Before Accepting Packets for Service Processing on page 67
- Postservice Filtering of Returning Service Traffic on page 68

Restrictions for Adaptive Services Interfaces

The following restrictions apply to adaptive services interfaces and service filters.

Adaptive Services Interfaces

You can apply a service filter to IPv4 or IPv6 traffic associated with a service set at an *adaptive services interface* only. Adaptive services interfaces are supported for the following hardware only:

- Adaptive Services (AS) PICs on M Series and T Series routers
- Multiservices (MS) PICs on M Series and T Series routers
- Multiservices (MS) DPCs on MX Series routers

System Logging to a Remote Host from M Series Routers

Logging of adaptive services interfaces messages to an external server by means of the **fxp0** or **em0** port is not supported on M Series routers. The architecture does not support system logging traffic out of a management interface. Instead, access to an external server is supported on a Packet Forwarding Engine interface.

Statement Hierarchy for Applying Service Filters

You can enable packet filtering of IPv4 or IPv6 traffic before a packet is accepted for input or output service processing. To do this, apply a service filter to the adaptive services interface input or output in conjunction with an interface service set.

You can also enable packet filtering of IPv4 or IPv6 traffic that is returning to the Packet Forwarding Engine after input service processing completes. To do this, apply a post-service filter to the adaptive services interface input.

The following configuration shows the hierarchy levels at which you can apply the service filters to adaptive services interfaces:

```
[edit]
interfaces {
  interface-name {
    unit unit-number {
      family (inet | inet6) {
        service {
          input {
```

```

        service-set service-set-name service-filter service-filter-name;
        post-service-filter service-filter-name;
    }
    output {
        service-set service-set-name service-filter service-filter-name;
    }
}
}
}
}
}
}
}

```

Associating Service Rules with Adaptive Services Interfaces

To define and group the service rules to be applied to an adaptive services interface, you define an *interface service set* by including the **service-set *service-set-name*** statement at the [edit services] hierarchy level.

To apply an interface service set to the input and output of an adaptive services interface, you include the **service-set *service-set-name*** at the following hierarchy levels:

- [edit interfaces *interface-name* unit *unit-number* input]
- [edit interfaces *interface-name* unit *unit-number* output]

If you apply a service set to one direction of an adaptive services interface but do not apply a service set to the other direction, an error occurs when you commit the configuration.

The adaptive services PIC performs different actions depending on whether the packet is sent to the PIC for input service or for output service. For example, you can configure a single service set to perform Network Address Translation (NAT) in one direction and destination NAT (dNAT) in the other direction.

Filtering Traffic Before Accepting Packets for Service Processing

To filter IPv4 or IPv6 traffic before accepting packets for input or output service processing, include the **service-set *service-set-name* service-filter *service-filter-name*** at one of the following interfaces:

- [edit interfaces *interface-name* unit *unit-number* family (inet | inet6) service input]
- [edit interfaces *interface-name* unit *unit-number* family (inet | inet6) service output]

For the **service-set-name**, specify a service set configured at the [edit services *service-set*] hierarchy level.

The service set retains the input interface information even after services are applied, so that functions such as filter-class forwarding and destination class usage (DCU) that depend on input interface information continue to work.

The following requirements apply to filtering inbound or outbound traffic before accepting packets for service processing:

- You configure the same service set on the input and output sides of the interface.

- If you include the **service-set** statement without an optional **service-filter** definition, the Junos OS assumes the match condition is true and selects the service set for processing automatically.
- The service filter is applied only if a service set is configured and selected.

You can include more than one service set definition on each side of an interface. The following guidelines apply:

- If you include multiple service sets, the router software evaluates them in the order in which they appear in the configuration. The system executes the first service set for which it finds a match in the service filter and ignores the subsequent definitions.
- A maximum of six service sets can be applied to an interface.
- When you apply multiple service sets to an interface, you must also configure and apply a service filter to the interface.

Postservice Filtering of Returning Service Traffic

As an option to filtering of IPv4 or IPv6 input service traffic, you can apply a service filter to IPv4 or IPv6 traffic that is returning to the services interface after the service set is executed. To apply a service filter in this manner, include the **post-service-filter service-filter-name** statement at the **[edit interfaces interface-name unit unit-number family (inet | inet6) service input]** hierarchy level.

Related Documentation

- Service Filter Overview on page 61
- How Service Filters Evaluate Packets on page 62
- Guidelines for Configuring Service Filters on page 64
- Example: Configuring and Applying Service Filters on page 211
- “Adaptive Services Overview” in the *Junos OS Services Interfaces Configuration Guide*
- “Configuring Service Sets to be Applied to Services Interfaces” in the *Junos OS Services Interfaces Configuration Guide*
- “Configuring Service Rules” in the *Junos OS Services Interfaces Configuration Guide*

CHAPTER 7

Introduction to Simple Filter Configuration

- Simple Filter Overview on page 69
- How Simple Filters Evaluate Packets on page 69
- Guidelines for Configuring Simple Filters on page 70
- Guidelines for Applying Simple Filters on page 74

Simple Filter Overview

Simple filters are supported on Gigabit Ethernet intelligent queuing 2 (IQ2) and Enhanced Queuing Dense Port Concentrator (DPC) interfaces only.

Simple filters are recommended for metropolitan Ethernet applications.

Related Documentation

- Stateless Firewall Filter Types on page 6
- How Simple Filters Evaluate Packets on page 69
- Guidelines for Configuring Simple Filters on page 70
- Guidelines for Applying Simple Filters on page 74
- Example: Configuring and Applying a Simple Filter on page 217

How Simple Filters Evaluate Packets

This topic covers the following information:

- Simple Filters That Contain a Single Term on page 69
- Simple Filters That Contain Multiple Terms on page 70
- Simple Filter Terms That Do Not Contain Any Match Conditions on page 70
- Simple Filter Terms That Do Not Contain Any Actions on page 70
- Simple Filter Default Action on page 70

Simple Filters That Contain a Single Term

For a simple filter that consists of a single term, the policy framework software evaluates a packet as follows:

- If the packet matches all the conditions, the actions are taken.

- If the packet matches all the conditions and no actions are specified, the packet is accepted.
- If the packet does not match all the conditions, it is discarded.

Simple Filters That Contain Multiple Terms

For a simple filter that consists of multiple terms, the policy framework software evaluates a packet against the terms in the filter sequentially, beginning with the first term in the filter, until either the packet matches all the conditions in one of the terms or there are no more terms in the filter.

- If the packet matches all the conditions in a term, the actions in that term are performed and evaluation of the packet ends at that term. Any subsequent terms in the filter are not used.
- If the packet does not match all the conditions in the term, evaluation of the packet proceeds to the next term in the filter.

Simple Filter Terms That Do Not Contain Any Match Conditions

For simple filters with a single term and for filters with multiple terms, if a term does not contain any match conditions, the actions are taken on any packet evaluated.

Simple Filter Terms That Do Not Contain Any Actions

If a simple filter term does not contain any actions, and if the packet matches the conditions in the term, the packet is accepted.

Simple Filter Default Action

Each simple filter has an *implicit discard* action at the end of the filter, which is equivalent to including the following example term **explicit_discard** as the final term in the simple filter:

```
term explicit_discard {  
    then discard;  
}
```

By default, if a packet matches none of the terms in a simple filter, the packet is discarded.

Related Documentation

- Simple Filter Overview on page 69
- Guidelines for Configuring Simple Filters on page 70
- Guidelines for Applying Simple Filters on page 74
- Example: Configuring and Applying a Simple Filter on page 217

Guidelines for Configuring Simple Filters

This topic covers the following information:

- Statement Hierarchy for Configuring Simple Filters on page 71
- Simple Filter Protocol Families on page 71
- Simple Filter Names on page 71

- Simple Filter Terms on page 71
- Simple Filter Match Conditions on page 72
- Simple Filter Terminating Actions on page 73
- Simple Filter Nonterminating Actions on page 73

Statement Hierarchy for Configuring Simple Filters

To configure a simple filter, include the **simple-filter *simple-filter-name*** statement at the **[edit firewall family inet]** hierarchy level.

```
[edit]
firewall {
  family inet {
    simple-filter simple-filter-name {
      term term-name {
        from {
          match-conditions;
        }
        then {
          actions;
        }
      }
    }
  }
}
```

Individual statements supported under the **simple-filter *simple-filter-name*** statement are described separately in this topic and are illustrated in the example of configuring and applying a simple filter.

Simple Filter Protocol Families

You can configure simple filters to filter IPv4 traffic (**family inet**) only. No other protocol family is supported for simple filters.

Simple Filter Names

Under the **family inet** statement, you can include **simple-filter *simple-filter-name*** statements to create and name simple filters. The filter name can contain letters, numbers, and hyphens (-) and be up to 64 characters long. To include spaces in the name, enclose the entire name in quotation marks (" ").

Simple Filter Terms

Under the **simple-filter *simple-filter-name*** statement, you can include **term *term-name*** statements to create and name filter terms.

- You must configure at least one term in a firewall filter.
- You must specify a unique name for each term within a firewall filter. The term name can contain letters, numbers, and hyphens (-) and can be up to 64 characters long. To include spaces in the name, enclose the entire name in quotation marks (" ").
- The order in which you specify terms within a firewall filter configuration is important. Firewall filter terms are evaluated in the order in which they are configured. By default, new terms are always added to the end of the existing filter. You can use the **insert** configuration mode command to reorder the terms of a firewall filter.

Simple filters do *not* support the **next term** action.

Simple Filter Match Conditions

Simple filter terms support only a subset of the IPv4 match conditions that are supported for standard stateless firewall filters.

Unlike standard stateless firewall filters, the following restrictions apply to simple filters:

- On MX Series routers with the Enhanced Queuing DPC, simple filters do *not* support the **forwarding-class** match condition.
- Simple filters support only one **source-address** and one **destination-address** prefix for each filter term. If you configure multiple prefixes, only the last one is used.
- Simple filters do *not* support multiple source addresses and destination addresses in a single term. If you configure multiple addresses, only the last one is used.
- Simple filters do *not* support negated match conditions, such as the **protocol-except** match condition or the **exception** keyword.
- Simple filters support a range of values for **source-port** and **destination-port** match conditions only. For example, you can configure **source-port 400-500** or **destination-port 600-700**.
- Simple filters do *not* support noncontiguous mask values.

Table 11 on page 72 lists the simple filter match conditions.

Table 11: Simple Filter Match Conditions

Match Condition	Description
destination-address <i>destination-address</i>	Match IP destination address.
destination-port <i>number</i>	<p>TCP or UDP destination port field.</p> <p>If you configure this match condition, we recommend that you also configure the protocol match statement to determine which protocol is being used on the port.</p> <p>In place of the numeric value, you can specify one of the following text aliases (the port numbers are also listed): afs (1483), bgp (179), biff (512), bootpc (68), bootps (67), cmd (514), cvspserver (2401), dhcp (67), domain (53), eklogin (2105), ekshell (2106), exec (512), finger (79), ftp (21), ftp-data (20), http (80), https (443), ident (113), imap (143), kerberos-sec (88), klogin (543), kpasswd (761), krb-prop (754), krbupdate (760), kshell (544), ldap (389), login (513), mobileip-agent (434), mobileip-mn (435), msdp (639), netbios-dgm (138), netbios-ns (137), netbios-ssn (139), nfsd (2049), nnntp (119), ntalk (518), ntp (123), pop3 (110), pptp (1723), printer (515), radacct (1813), radius (1812), rip (520), rinit (2108), smtp (25), snmp (161), snmptrap (162), snpp (444), socks (1080), ssh (22), sunrpc (111), syslog (514), tacacs-ds (65), talk (517), telnet (23), tftp (69), timed (525), who (513), or xdmcp (177).</p>
forwarding-class <i>class</i>	<p>Match the forwarding class of the packet.</p> <p>Specify assured-forwarding, best-effort, expedited-forwarding, or network-control.</p> <p>For information about forwarding classes and router-internal output queues, see the Junos OS Class of Service Configuration Guide.</p>

Table 11: Simple Filter Match Conditions (*continued*)

Match Condition	Description
protocol number	IP protocol field. In place of the numeric value, you can specify one of the following text aliases (the field values are also listed): ah (51), dstopts (60), egp (8), esp (50), fragment (44), gre (47), hop-by-hop (0), icmp (1), icmp6 (58), icmpv6 (58), igmp (2), ipip (4), ipv6 (41), ospf (89), pim (103), rsvp (46), sctp (132), tcp (6), udp (17), or vrrp (112).
source-address <i>ip-source-address</i>	Match the IP source address.
source-port number	Match the UDP or TCP source port field. If you configure this match condition, we recommend that you also configure the protocol match statement to determine which protocol is being used on the port. In place of the numeric field, you can specify one of the text aliases listed for destination-port .

Simple Filter Terminating Actions

Simple filters do *not* support explicitly configurable terminating actions, such as **accept**, **reject**, and **discard**. Terms configured in a simple filter always accept packets.

Simple filters do *not* support the **next** action.

Simple Filter Nonterminating Actions

Simple filters support only the following nonterminating actions:

- **forwarding-class** (*forwarding-class* | **assured-forwarding** | **best-effort** | **expedited-forwarding** | **network-control**)



NOTE: On the MX Series routers with the Enhanced Queuing DPC, the forwarding class is not supported as a from match condition.

- **loss-priority** (**high** | **low** | **medium-high** | **medium-low**)

Simple filters do not support actions that perform other functions on a packet (such as incrementing a counter, logging information about the packet header, sampling the packet data, or sending information to a remote host using the system log functionality).

Related Documentation

- Simple Filter Overview on page 69
- How Simple Filters Evaluate Packets on page 69
- Guidelines for Applying Simple Filters on page 74
- Example: Configuring and Applying a Simple Filter on page 217

Guidelines for Applying Simple Filters

This topic covers the following information:

- Statement Hierarchy for Applying Simple Filters on page 74
- Restrictions for Applying Simple Filters on page 74

Statement Hierarchy for Applying Simple Filters

You can apply a simple filter to the IPv4 ingress traffic at a logical interface by including the **simple-filter** input *simple-filter-name* statement at the **[edit interfaces *interface-name* unit *unit-number* family inet]** hierarchy level.

```
[edit]
interfaces {
  interface-name {
    unit logical-unit-number {
      family inet {
        simple-filter {
          input filter-name;
        }
      }
    }
  }
}
```

Restrictions for Applying Simple Filters

You can apply a simple filter to the ingress IPv4 traffic at a logical interface configured on the following hardware only:

- Gigabit Ethernet intelligent queuing (IQ2) PICs installed on M120, M320, or T Series routers.
- Enhanced Queuing Dense Port Concentrators (EQ DPCs) installed on MX Series routers.

For more information about Ethernet IQ2 PICs and EQ DPCs and related features, see the [Junos OS Class of Service Configuration Guide](#). For more information about configuring the MX Series routers, on which EQ DPCs are supported, see the [Junos OS Layer 2 Configuration Guide](#).

The following additional restrictions pertain to applying simple filters:

- Simple filters are not supported on Modular Port Concentrator (MPC) interfaces, including Enhanced Queuing MPC interfaces.
- Simple filters are not supported for interfaces in an aggregated-Ethernet bundle.
- You can apply simple filters to **family inet** traffic only. No other protocol family is supported.
- You can apply simple filters to ingress traffic only. Egress traffic is not supported.
- You can apply only a single simple filter to a supported logical interface. Input lists are not supported.

**Related
Documentation**

- Simple Filter Overview on page 69
- How Simple Filters Evaluate Packets on page 69
- Guidelines for Configuring Simple Filters on page 70
- Example: Configuring and Applying a Simple Filter on page 217

CHAPTER 8

Introduction to Firewall Filter Configuration in Logical Systems

- Stateless Firewall Filters in Logical Systems Overview on page 77
- Guidelines for Configuring and Applying Firewall Filters in Logical Systems on page 78
- References from a Firewall Filter in a Logical System to Subordinate Objects on page 81
- References from a Firewall Filter in a Logical System to Nonfirewall Objects on page 82
- References from a Nonfirewall Object in a Logical System to a Firewall Filter on page 84

Stateless Firewall Filters in Logical Systems Overview

This topic covers the following information:

- Logical Systems on page 77
- Stateless Firewall Filters in Logical Systems on page 77
- Identifiers for Firewall Objects in Logical Systems on page 77

Logical Systems

With the Junos OS, you can partition a single physical router into multiple logical devices that perform independent routing tasks. Because logical systems perform a subset of the tasks once handled by the physical router, logical systems offer an effective way to maximize the use of a single router.

Stateless Firewall Filters in Logical Systems

You can configure a separate set of stateless firewall filters for each logical system on a router. To configure a filter in a logical system, you must define the filter in the **firewall** stanza at the **[edit logical-systems logical-system-name]** hierarchy level, and you must apply the filter to a logical interface that is also configured at the **[edit logical-systems logical-system-name]** hierarchy level.

Identifiers for Firewall Objects in Logical Systems

To identify firewall objects configured under logical systems, operational **show** commands and firewall-related SNMP MIB objects include a **__logical-system-name/** prefix in the object name. For example, firewall objects configured under the **ls1** logical system include **__ls1/** as the prefix.

Related Documentation

- Stateless Firewall Filter Types on page 6
- Guidelines for Configuring and Applying Firewall Filters in Logical Systems on page 78
- Unsupported Firewall Filter Statements for Logical Systems on page 283
- Unsupported Actions for Firewall Filters in Logical Systems on page 285
- Example: Configuring a Stateless Firewall Filter to Protect a Logical System Against ICMP Floods on page 223
- “Introduction to Logical Systems” in the *Junos OS Logical Systems Configuration Guide*
- “Logical Systems Operations and Restrictions” in the *Junos OS Logical Systems Configuration Guide*

Guidelines for Configuring and Applying Firewall Filters in Logical Systems

This topic covers the following information:

- Statement Hierarchy for Configuring Firewall Filters in Logical Systems on page 78
- Filter Types in Logical Systems on page 79
- Firewall Filter Protocol Families in Logical Systems on page 79
- Firewall Filter Match Conditions in Logical Systems on page 79
- Firewall Filter Actions in Logical Systems on page 80
- Statement Hierarchy for Applying Firewall Filters in Logical Systems on page 80

Statement Hierarchy for Configuring Firewall Filters in Logical Systems

To configure a firewall filter in a logical system, include the **filter**, **service-filter**, or **simple-filter** statement at the **[edit logical-systems *logical-system-name* firewall family *family-name*]** hierarchy level.

```
[edit]
logical systems {
  logical-system-name {
    firewall {
      family family-name {
        filter filter-name {
          interface-specific;
          physical-interface-filter;
          term term-name {
            filter filter-name;
            from {
              match-conditions;
            }
            then {
              actions;
            }
          }
        }
      }
      service-filter filter-name { # For 'family inet' or 'family inet6' only.
        term term-name {
          from {
```


Filter Types in Logical Systems

In a logical system, you can use the same types of stateless firewall filters that are available on a physical router:

- ## Firewall Filter Protocol Families in Logical Systems

In a logical system, you can filter the same protocol families as you can on a physical router.

- ## Firewall Filter Match Conditions in Logical Systems

79

Firewall Filter Actions in Logical Systems

There are no special restrictions on the actions supported with stateless firewall filters in logical systems.

Statement Hierarchy for Applying Firewall Filters in Logical Systems

To apply a firewall filter in a logical system, include the **filter** *filter-name*, **service-filter** *service-filter-name*, or **simple-filter** *simple-filter-name* statement to a logical interface in the logical system.

The following configuration shows the hierarchy levels at which you can apply the statements:

```
[edit]
logical-systems logical-system-name {
  interfaces {
    interface-name {
      unit logical-unit-number {
        family family-name {
          filter {
            group group-name;
            input filter-name;
            input-list [ filter-names ];
            output filter-name;
            output-list [ filter-names ]
          }
          rpf-check { # For 'family inet' or 'family inet6' only.
            fail-filter filter-name;
            mode loose;
          }
          service { # For 'family inet' or 'family inet6' only.
            input {
              service-set service-set-name <service-filter service-filter-name>;
              post-service-filter service-filter-name;
            }
            output {
              service-set service-set-name <service-filter service-filter-name>;
            }
          }
          simple-filter { # For 'family inet' only.
            input simple-filter-name;
          }
        }
      }
    }
  }
}
```

Related Documentation

- Stateless Firewall Filters in Logical Systems Overview on page 77
- References from a Firewall Filter in a Logical System to Subordinate Objects on page 81
- References from a Firewall Filter in a Logical System to Nonfirewall Objects on page 82
- References from a Nonfirewall Object in a Logical System to a Firewall Filter on page 84

- Example: Configuring a Stateless Firewall Filter to Protect a Logical System Against ICMP Floods on page 223
- Unsupported Firewall Filter Statements for Logical Systems on page 283
- Unsupported Actions for Firewall Filters in Logical Systems on page 285

References from a Firewall Filter in a Logical System to Subordinate Objects

This topic covers the following information:

- Resolution of References from a Firewall Filter to Subordinate Objects on page 81
- Valid Reference from a Firewall Filter to a Subordinate Object on page 81

Resolution of References from a Firewall Filter to Subordinate Objects

If a firewall filter defined in a logical system references a subordinate object (for example, a policer or prefix list), that subordinate object must be defined within the **firewall** stanza of the same logical system. For example, if a firewall filter configuration references a policer, the firewall filter and the policer must be configured under the same **[edit logical-systems logical-system-name firewall]** hierarchy level.

This rule applies even if the same policer is configured under the main firewall configuration or if the same policer is configured as part of a firewall in another logical system.

Valid Reference from a Firewall Filter to a Subordinate Object

In this example, the firewall filter **filter1** references the policer **pol1**. Both **filter1** and **pol1** are defined under the same firewall object. This configuration is valid. If **pol1** had been defined under another firewall object, the configuration would not be valid.

```
[edit]
logical systems {
  ls-A {
    firewall {
      policer pol1 {
        if-exceeding {
          bandwidth-limit 401k;
          burst-size-limit 50k;
        }
        then discard;
      }
      filter filter1 {
        term one {
          from {
            source-address 12.1.0.0/16;
          }
          then {
            reject host-unknown;
          }
        }
        term two {
          from {
            source-address 12.2.0.0/16;
```

```
        }
        then policer pol1;
    }
}
}
```

**Related
Documentation**

- Stateless Firewall Filters in Logical Systems Overview on page 77
- Guidelines for Configuring and Applying Firewall Filters in Logical Systems on page 78
- References from a Firewall Filter in a Logical System to Nonfirewall Objects on page 82
- References from a Nonfirewall Object in a Logical System to a Firewall Filter on page 84

References from a Firewall Filter in a Logical System to Nonfirewall Objects

This topic covers the following information:

- Resolution of References from a Firewall Filter to Nonfirewall Objects on page 82
- Valid Reference to a Nonfirewall Object Outside of the Logical System on page 82

Resolution of References from a Firewall Filter to Nonfirewall Objects

In many cases, a firewall configuration references objects outside the firewall configuration. As a general rule, the referenced object must be defined under the same logical system as the referencing object. However, there are cases when the configuration of the referenced object is not supported at the `[edit logical-systems logical-system-name]` hierarchy level.

Valid Reference to a Nonfirewall Object Outside of the Logical System

This example configuration illustrates an exception to the general rule that the objects referenced by a firewall filter in a logical system must be defined under the same logical system as the referencing object.

In the following scenario, the service filter `inetsf1` is applied to IPv4 traffic associated with the service set `fred` at the logical interface `fe-0/3/2.0`, which is on an adaptive services interface.

- Service filter `inetsf1` is defined in `ls-B` and references prefix list `prefix1`.
- Service set `fred` is defined at the main services hierarchy level, and the policy framework software searches the `[edit services]` hierarchy for the definition of the `fred` service set.

Because service rules cannot be configured in logical systems, firewall filter configurations in the `[edit logical-systems logical-system logical-system-name]` hierarchy are allowed to reference *service sets* outside the logical system hierarchy.

```
[edit]
logical-systems {
  ls-B {
    interfaces {
      fe-0/3/2 {
        unit 0 {
          family inet {
```

```
        service {
            input {
                service-set fred service-filter inetsf1;
            }
        }
    }
}
policy-options {
    prefix-list prefix1 {
        1.1.0.0/16;
        1.2.0.0/16;
        1.3.0.0/16;
    }
}
firewall { # Under logical-system 'ls-B'.
    family inet {
        filter filter1 {
            term one {
                from {
                    source-address {
                        12.1.0.0/16;
                    }
                }
                then {
                    reject host-unknown;
                }
            }
            term two {
                from {
                    source-address {
                        12.2.0.0/16;
                    }
                }
                then policer pol1;
            }
        }
        service-filter inetsf1 {
            term term1 {
                from {
                    source-prefix-list {
                        prefix1;
                    }
                }
                then count prefix1;
            }
        }
    }
    policer pol1 {
        if-exceeding {
            bandwidth-limit 401k;
            burst-size-limit 50k;
        }
        then discard;
    }
}
```

```
    }  
  }  
} # End of logical systems configuration.  
services { # Main services hierarchy level.  
  service-set fred {  
    max-flows 100;  
    interface-service {  
      service-interface sp-1/2/0.0;  
    }  
  }  
}
```

**Related
Documentation**

- Stateless Firewall Filters in Logical Systems Overview on page 77
- Guidelines for Configuring and Applying Firewall Filters in Logical Systems on page 78
- References from a Firewall Filter in a Logical System to Subordinate Objects on page 81
- References from a Nonfirewall Object in a Logical System to a Firewall Filter on page 84

References from a Nonfirewall Object in a Logical System to a Firewall Filter

This topic covers the following information:

- Resolution of References from a Nonfirewall Object to a Firewall Filter on page 84
- Invalid Reference to a Firewall Filter Outside of the Logical System on page 85
- Valid Reference to a Firewall Filter Within the Logical System on page 86
- Valid Reference to a Firewall Filter Outside of the Logical System on page 88

Resolution of References from a Nonfirewall Object to a Firewall Filter

If a nonfirewall filter object in a logical system references an object in a firewall filter configured in a logical system, the reference is resolved using the following logic:

- If the nonfirewall filter object is configured in a logical system that includes firewall filter configuration statements, the policy framework software searches the **[edit logical-systems *logical-system-name* firewall]** hierarchy level. Firewall filter configurations that belong to *other* logical systems or to the main **[edit firewall]** hierarchy level are not searched.
- If the nonfirewall filter object is configured in a logical system that does not include any firewall filter configuration statements, the policy framework software searches the firewall configurations defined at the **[edit firewall]** hierarchy level.

Invalid Reference to a Firewall Filter Outside of the Logical System

This example configuration illustrates an unresolvable reference from a nonfirewall object in a logical system to a firewall filter.

In the following scenario, the stateless firewall filters **filter1** and **fred** are applied to the logical interface **fe-0/3/2.0** in the logical system **ls-C**.

- Filter **filter1** is defined in **ls-C**.
- Filter **fred** is defined in the main firewall configuration.

Because **ls-C** contains firewall filter statements (for **filter1**), the policy framework software resolves references to and from firewall filters by searching the **[edit logical systems ls-C firewall]** hierarchy level. Consequently, the reference from **fe-0/3/2.0** in the logical system to **fred** in the main firewall configuration cannot be resolved.

```
[edit]
logical-systems {
  ls-C {
    interfaces {
      fe-0/3/2 {
        unit 0 {
          family inet {
            filter {
              input-list [ filter1 fred ];
            }
          }
        }
      }
    }
  }
}
firewall { # Under logical system 'ls-C'.
  family inet {
    filter filter1 {
      term one {
        from {
          source-address 12.1.0.0/16;
        }
        then {
          reject host-unknown;
        }
      }
      term two {
        from {
          source-address 12.2.0.0/16;
        }
        then policer pol1;
      }
    }
  }
  policer pol1 {
    if-exceeding {
      bandwidth-limit 401k;
      burst-size-limit 50k;
    }
    then discard;
  }
}
```

```

    }
  }
} # End of logical systems
firewall { # Under the main firewall hierarchy level
  family inet {
    filter fred {
      term one {
        from {
          source-address 11.1.0.0/16;
        }
        then {
          log;
          reject host-unknown;
        }
      }
    }
  }
} # End of main firewall configurations.

```

Valid Reference to a Firewall Filter Within the Logical System

This example configuration illustrates resolvable references from a nonfirewall object in a logical system to two firewall filter.

In the following scenario, the stateless firewall filters **filter1** and **fred** are applied to the logical interface **fe-0/3/2.0** in the logical system **ls-C**.

- Filter **filter1** is defined in **ls-C**.
- Filter **fred** is defined in **ls-C** and also in the main firewall configuration.

Because **ls-C** contains firewall filter statements, the policy framework software resolves references to and from firewall filters by searching the **[edit logical systems ls-C firewall]** hierarchy level. Consequently, the references from **fe-0/3/2.0** in the logical system to **filter1** and **fred** use the stateless firewall filters configured in **ls-C**.

```

[edit]
logical-systems {
  ls-C {
    interfaces {
      fe-0/3/2 {
        unit 0 {
          family inet {
            filter {
              input-list [ filter1 fred ];
            }
          }
        }
      }
    }
  }
  firewall { # Under logical system 'ls-C'.
    family inet {
      filter filter1 {
        term one {
          from {
            source-address 12.1.0.0/16;
          }
          then {

```



```
        reject host-unknown;
    }
}
term two {
    from {
        source-address 12.2.0.0/16;
    }
    then policer pol1;
}
}
filter fred { # This 'fred' is in 'ls-C'.
    term one {
        from {
            source-address 10.1.0.0/16;
        }
        then {
            log;
            reject host-unknown;
        }
    }
}
}
policer pol1 {
    if-exceeding {
        bandwidth-limit 401k;
        burst-size-limit 50k;
    }
    then discard;
}
}
}
} # End of logical systems configurations.
firewall { # Main firewall filter hierarchy level
    family inet {
        filter fred {
            term one {
                from {
                    source-address 11.1.0.0/16;
                }
                then {
                    log;
                    reject host-unknown;
                }
            }
        }
    }
}
} # End of main firewall configurations.
```

Valid Reference to a Firewall Filter Outside of the Logical System

This example configuration illustrates resolvable references from a nonfirewall object in a logical system to two firewall filter.

In the following scenario, the stateless firewall filters **filter1** and **fred** are applied to the logical interface **fe-0/3/2.0** in the logical system **ls-C**.

- Filter **filter1** is defined in the main firewall configuration.
- Filter **fred** is defined in the main firewall configuration.

Because **ls-C** does not contain any firewall filter statements, the policy framework software resolves references to and from firewall filters by searching the **[edit firewall]** hierarchy level. Consequently, the references from **fe-0/3/2.0** in the logical system to **filter1** and **fred** use the stateless firewall filters configured in the main firewall configuration.

```
[edit]
logical-systems {
  ls-C {
    interfaces {
      fe-0/3/2 {
        unit 0 {
          family inet {
            filter {
              input-list [ filter1 fred ];
            }
          }
        }
      }
    }
  }
}
} # End of logical systems configurations.
firewall { # Main firewall hierarchy level.
  family inet {
    filter filter1 {
      term one {
        from {
          source-address 12.1.0.0/16;
        }
        then {
          reject host-unknown;
        }
      }
      term two {
        from {
          source-address 12.2.0.0/16;
        }
        then policer pol1;
      }
    }
  }
  filter fred {
    term one {
      from {
        source-address 11.1.0.0/16;
      }
      then {
```

- **Related Documentation**
- Stateless Firewall Filters in Logical Systems Overview on page 77
- Guidelines for Configuring and Applying Firewall Filters in Logical Systems on page 78
- References from a Firewall Filter in a Logical System to Subordinate Objects on page 81
- References from a Firewall Filter in a Logical System to Nonfirewall Objects on page 82

PART 2

Configuration

- Standard Firewall Filter Configurations That Match Packets on page 93
- Standard Firewall Filters That Count Packets on page 105
- Standard Firewall Filters That Act on Packets on page 117
- Standard Firewall Filters for Trusted Sources on page 125
- Standard Firewall Filters for Flood Prevention on page 151
- Standard Firewall Filters for Fragment Handling on page 161
- Standard Firewall Filters for Setting Rate Limits on page 167
- Standard Firewall Configuration on page 171
- Standard Firewall Configuration Options on page 201
- Service Filter Configuration on page 211
- Simple Filter Configuration on page 217
- Firewall Filter Configuration in Logical Systems on page 223

CHAPTER 9

Standard Firewall Filter Configurations That Match Packets

- Example: Configuring a Filter to Match on IPv6 Flags on page 93
- Example: Configuring a Filter to Match on Port and Protocol Fields on page 94
- Example: Configuring a Filter to Match on Two Unrelated Criteria on page 97
- Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List on page 100

Example: Configuring a Filter to Match on IPv6 Flags

This example shows how to configure a filter to match on IPv6 TCP flags.

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

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you configure a filter to match on IPv6 TCP flags. You can use this example to configure IPv6 TCP flags in the SRX100, SRX210, SRX240, SRX650, and J Series security devices and in M Series, MX Series, and T Series routing devices.

Configuration

Step-by-Step Procedure

To configure a filter to match on IPv6 TCP flags:

1. Include the family statement at the firewall hierarchy level, specifying **inet6** as the protocol family.

[edit]

user@host# edit firewall family inet6

2. Create the stateless firewall filter.

```
[edit firewall family inet6]  
user@host# edit filter tcpfilt
```

3. Define the first term for the filter.

```
[edit firewall family inet6 filter tcpfilt]  
user@host# edit term 1
```

4. Define the source address match conditions for the term.

```
[edit firewall family inet6 filter tcpfilt term 1]  
user@host# set from next-header tcp tcp-flags syn
```

5. Define the actions for the term.

```
[edit firewall family inet6 filter tcpfilt term 1]  
user@host# set then count tcp_syn_pkt log accept
```

6. If you are done configuring the device, commit the configuration.

```
[edit firewall family inet6 filter tcpfilt term 1]  
user@host# top
```

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

Verification

To confirm that the configuration is working properly, enter the **show firewall filter tcpfilt** command.

Example: Configuring a Filter to Match on Port and Protocol Fields

This example shows how to configure a standard stateless firewall filter to match on destination port and protocol fields.

- Requirements on page 94
- Overview on page 94
- Configuration on page 95
- Verification on page 97

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you configure a stateless firewall filter that accepts all IPv4 packets except for TCP and UDP packets. TCP and UDP packets are accepted if destined for the SSH port or the Telnet port. All other packets are rejected.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

- Configure the Stateless Firewall Filter on page 95
- Apply the Stateless Firewall Filter to a Logical Interface on page 96
- Confirm and Commit Your Candidate Configuration on page 96

CLI Quick Configuration To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level:

```
set firewall family inet filter filter1 term term1 from protocol-except tcp
set firewall family inet filter filter1 term term1 from protocol-except udp
set firewall family inet filter filter1 term term1 then accept
set firewall family inet filter filter1 term term2 from address 192.168.0.0/16
set firewall family inet filter filter1 term term2 then reject
set firewall family inet filter filter1 term term3 from destination-port ssh
set firewall family inet filter filter1 term term3 from destination-port telnet
set firewall family inet filter filter1 term term3 then accept
set firewall family inet filter filter1 term term4 then reject
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.3/30
set interfaces ge-0/0/1 unit 0 family inet filter input filter1
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure To configure the stateless firewall filter **filter1**:

1. Create the IPv4 stateless firewall filter.

[edit]
user@host# edit firewall family inet filter filter1
2. Configure a term to accept all traffic except for TCP and UDP packets.

[edit firewall family inet filter filter1]
user@host# set term term1 from protocol-except tcp
user@host# set term term1 from protocol-except udp
user@host# set term term1 then accept
3. Configure a term to reject packets to or from the 192.168/16 prefix.

[edit firewall family inet filter filter1]
user@host# set term term2 from address 192.168.0.0/16
user@host# set term term2 then reject
4. Configure a term to accept packets destined for either the SSH port or the Telnet port.

[edit firewall family inet filter filter1]
user@host# set term term3 from destination-port ssh
user@host# set term term3 from destination-port telnet
user@host# set term term3 then accept

5. Configure the last term to reject all packets.

```
[edit firewall family inet filter filter1]
user@host# set term term4 then reject
```

Apply the Stateless Firewall Filter to a Logical Interface

Step-by-Step Procedure

To apply the stateless firewall filter to a logical interface:

1. Configure the logical interface to which you will apply the stateless firewall filter.

```
[edit]
user@host# edit interfaces ge-0/0/1 unit 0 family inet
```

2. Configure the interface address for the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set address 10.1.2.3/30
```

3. Apply the stateless firewall filter to the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set filter input filter1
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter filter1 {
    term term1 {
      from {
        protocol-except [tcp udp];
      }
      then {
        accept;
      }
    }
    term term2 {
      from {
        address 192.168/16;
      }
      then {
        reject;
      }
    }
    term term3 {
      from {
        destination-port [ssh telnet];
      }
      then {
```

```

        accept;
    }
}
term term4 {
    then {
        reject;
    }
}
}
}

```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            filter {
                input filter1;
            }
            address 10.1.2.3/30;
        }
    }
}

```

3. If you are done configuring the device, commit your candidate configuration.

```

[edit]
user@host# commit

```

Verification

To confirm that the configuration is working properly, enter the **show firewall filter filter1** operational mode command.

Related Documentation

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Filter to Match on IPv6 Flags on page 93
- Example: Configuring a Filter to Match on Two Unrelated Criteria on page 97

Example: Configuring a Filter to Match on Two Unrelated Criteria

This example shows how to configure a standard stateless firewall filter to match on two unrelated criteria.

- Requirements on page 98
- Overview on page 98
- Configuration on page 98
- Verification on page 100

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you use a standard stateless firewall filter to match IPv4 packets that are either OSPF packets or packets that come from an address in the prefix 10.108/16, and send an **administratively-prohibited** ICMP message for all packets that do not match.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configuring the IPv4 Firewall Filter on page 98
- Applying the IPv4 Firewall Filter to a Logical Interface on page 99

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter ospf_or_131 term protocol_match from protocol ospf
set firewall family inet filter ospf_or_131 term address-match from source-address
  10.108.0.0/16
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.3/30
set interfaces ge-0/0/1 unit 0 family inet filter input ospf_or_131
```

Configuring the IPv4 Firewall Filter

Step-by-Step Procedure

To configure the IPv4 firewall filter:

1. Enable configuration of the IPv4 firewall filter.

```
[edit]
user@host# edit firewall family inet filter ospf_or_131
```

2. Configure the first term to accept OSPF packets.

```
[edit firewall family inet filter ospf_or_131]
user@host# set term protocol_match from protocol ospf
```

Packets that match the condition are accepted by default. Because another term follows this term, packets that do not match this condition are evaluated by the next term.

3. Configure the second term to accept packets from any IPv4 address in a particular prefix.

```
[edit firewall family inet filter ospf_or_131]
user@host# set term address_match from source-address 10.108.0.0/16
```

Packets that match this condition are accepted by default. Because this is the last term in the filter, packets that do not match this condition are discarded by default.

Results Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter ospf_or_131 {
    term protocol_match {
      from {
        protocol ospf;
      }
    }
    term address_match {
      from {
        source-address {
          10.108.0.0/16;
        }
      }
    }
  }
}
```

Applying the IPv4 Firewall Filter to a Logical Interface

Step-by-Step Procedure To apply the stateless firewall filter to a logical interface:

1. Enable configuration of a logical interface.

```
[edit]
user@host# edit interfaces ge-0/0/1 unit 0 family inet
```

2. Configure an IP address for the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set address 10.1.2.3/30
```

3. Apply the IPv4 firewall filter to the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set filter input ospf_or_131
```

Results Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
  unit 0 {
    family inet {
      filter {
        input ospf_or_131;
      }
    }
  }
}
```

```
        }  
        address 10.1.2.3/30;  
    }  
}
```

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

Verification

To confirm that the configuration is working properly, enter the **show firewall filter ospf_or_131** operational mode command.

Related Documentation

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Filter to Match on IPv6 Flags on page 93
- Example: Configuring a Filter to Match on Port and Protocol Fields on page 94

Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List

This example shows how to configure a standard stateless firewall filter that limits certain TCP and ICMP traffic destined for the Routing Engine:

- Requirements on page 100
- Overview on page 100
- Configuration on page 101
- Verification on page 103

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you create a stateless firewall filter that blocks all TCP connection attempts to port **179** from all requesters except the specified BGP peers.

The source prefix list **plist_bgp179** specifies the list of source prefixes that contain allowed BGP peers.

The stateless firewall filter **filter_bgp179** matches all packets from the source prefix list **plist_bgp179** to the destination port number **179**.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

- Configure the Source Prefix List on page 101
- Configure the Stateless Firewall Filter on page 101
- Apply the Firewall Filter to the Loopback Interface on page 102
- Confirm and Commit Your Candidate Configuration on page 102

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options prefix-list plist_bgp179 apply-path "protocols bgp group <*> neighbor <*>"
set firewall family inet filter filter_bgp179 term 1 from source-address 0.0.0.0/0
set firewall family inet filter filter_bgp179 term 1 from source-prefix-list plist_bgp179 except
set firewall family inet filter filter_bgp179 term 1 from destination-port bgp
set firewall family inet filter filter_bgp179 term 1 then reject
set firewall family inet filter filter_bgp179 term 2 then accept
set interfaces lo0 unit 0 family inet filter input filter_bgp179
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

Configure the Source Prefix List

Step-by-Step Procedure

To configure **plist_bgp179**, the list of source prefixes that contain allowed BGP peers:

- Expand the prefix list **bgp179** to include all prefixes pointed to by the BGP peer group defined by **protocols bgp group <*> neighbor <*>**.

```
[edit]
set policy-options prefix-list plist_bgp179 apply-path "protocols bgp group <*> neighbor <*>"
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter that blocks all TCP connection attempts to port 179 from all requestors except specified BGP peers:

1. Create the stateless firewall filter **filter_bgp179**.

```
[edit]
user@host# edit firewall family inet filter filter_bgp179
```

2. Define the filter term that rejects TCP connection attempts to port 179 from all requestors except the specified BGP peers.

```
[edit firewall family inet filter filter_bgp179]
user@host# set term term1 from source-address 0.0.0.0/0
user@host# set term term1 from source-prefix-list bgp179 except
user@host# set term term1 from destination-port bgp
user@host# set term term1 then reject
```

3. Define the other filter term to accept all packets.

```
[edit firewall family inet filter filter_bgp179]
user@host# set term term2 then accept
```

Apply the Firewall Filter to the Loopback Interface

Step-by-Step Procedure

- To apply the firewall filter to the loopback interface:

```
[edit]
user@host# set interfaces lo0 unit 0 family inet filter input filter_bgp179
user@host# set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the source prefix list by entering the **show policy-options** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show policy-options
prefix-list plist_bgp179 {
    apply-path "protocols bgp group <*> neighbor <*>";
}
```

2. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
    filter filter_bgp179 {
        term 1 {
            from {
                source-address {
                    0.0.0.0/0;
                }
                source-prefix-list {
                    plist_bgp179 except;
                }
                destination-port bgp;
            }
            then {
                reject;
            }
        }
        term 2 {
            then {
                accept;
            }
        }
    }
}
```


3. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
lo0 {
  unit 0 {
    family inet {
      filter {
        input filter_bgp179;
      }
      address 127.0.0.1/32;
    }
  }
}
```

4. When you are done configuring the device, commit your candidate configuration.

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

Verification

Confirm that the configuration is working properly.

Displaying the Firewall Filter Applied to the Loopback Interface

Purpose Verify that the firewall filter **filter_bgp179** is applied to the IPv4 input traffic at the logical interface **lo0.0**.

Action Use the **show interfaces statistics** operational mode command for logical interface **lo0.0**, and include the **detail** option. Under the **Protocol inet** section of the command output section, the **Input Filters** field displays the name of the stateless firewall filter applied to the logical interface in the input direction:

```
[edit]
user@host> show interfaces statistics lo0.0 detail
Logical interface lo0.0 (Index 321) (SNMP ifIndex 16) (Generation 130)
  Flags: SNMP-Traps Encapsulation: Unspecified
  Traffic statistics:
    Input bytes : 0
    Output bytes : 0
    Input packets: 0
    Output packets: 0
  Local statistics:
    Input bytes : 0
    Output bytes : 0
    Input packets: 0
    Output packets: 0
  Transit statistics:
    Input bytes : 0 0 bps
    Output bytes : 0 0 bps
    Input packets: 0 0 pps
    Output packets: 0 0 pps
  Protocol inet, MTU: Unlimited, Generation: 145, Route table: 0
  Flags: Sendbroadcast-pkt-to-re
```

Input Filters: filter_bgp179
Addresses, Flags: Primary
Destination: Unspecified, Local: 127.0.0.1, Broadcast: Unspecified,
Generation: 138

**Related
Documentation**

- Basic Uses for Standard Firewall Filters on page 24
- Firewall Filter Match Conditions Based on Address Fields on page 32
- Example: Configuring a Stateless Firewall Filter to Protect Against TCP and ICMP Floods on page 151
- Example: Configuring a Filter to Accept Packets Based on IPv6 TCP Flags on page 158
- “**prefix-list on page 307**” in the *Junos OS Routing Policy Configuration Guide*

CHAPTER 10

Standard Firewall Filters That Count Packets

- Example: Configuring a Filter to Count Accepted and Rejected Packets on page 105
- Example: Configuring a Filter to Count and Discard IP Options Packets on page 108
- Example: Configuring a Filter to Count IP Options Packets on page 111

Example: Configuring a Filter to Count Accepted and Rejected Packets

This example shows how to configure a firewall filter to count packets.

- Requirements on page 105
- Overview on page 105
- Configuration on page 106
- Verification on page 108

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you use a stateless firewall filter to reject all addresses except 192.168.5.0/24.

Topology

In the first term, the match condition **address 192.168.5.0/24 except** causes this address to be considered a mismatch, and this address is passed to the next term in the filter. The match condition **address 0.0.0.0/0** matches all other packets, and these are counted, logged, and rejected.

In the second term, all packets that passed through the first term (that is, packets whose address matches 192.168.5.0/24) are counted, logged, and accepted.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configure the Stateless Firewall Filter on page 106
- Apply the Stateless Firewall Filter to a Logical Interface on page 107
- Confirm and Commit Your Candidate Configuration on page 107

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter fire1 term 1 from address 192.168.5.0/24 except
set firewall family inet filter fire1 term 1 from address 0.0.0.0/0
set firewall family inet filter fire1 term 1 then count reject_pref1_1
set firewall family inet filter fire1 term 1 then log
set firewall family inet filter fire1 term 1 then reject
set firewall family inet filter fire1 term 2 then count reject_pref1_2
set firewall family inet filter fire1 term 2 then log
set firewall family inet filter fire1 term 2 then accept
set interfaces ge-0/0/1 unit 0 family inet filter input fire1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.3/30
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter **fire1**:

1. Create the stateless firewall filter **fire1**.

```
[edit]
user@host# edit firewall family inet filter fire1
```

2. Configure the first term to reject all addresses except those to or from the **192.168.5.0/24** prefix and then count, log, and reject all other packets.

```
[edit firewall family inet filter fire1]
user@host# set term 1 from address 192.168.5.0/24 except
user@host# set term 1 from address 0.0.0.0/0
user@host# set term 1 then count reject_pref1_1
user@host# set term 1 then log
user@host# set term 1 then reject
```

3. Configure the next term to count, log, and accept packets in the **192.168.5.0/24** prefix.

```
[edit firewall family inet filter fire1]
user@host# set term 2 then count reject_pref1_2
user@host# set term 2 then log
user@host# set term 2 then accept
```

Apply the Stateless Firewall Filter to a Logical Interface

Step-by-Step Procedure

To apply the stateless firewall filter to a logical interface:

1. Configure the logical interface to which you will apply the stateless firewall filter.

```
[edit]
user@host# edit interfaces ge-0/0/1 unit 0 family inet
```

2. Configure the interface address for the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set address 10.1.2.3/30
```

3. Apply the stateless firewall filter to the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set filter input fire1
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter fire1 {
    term 1 {
      from {
        address {
          192.168.5.0/24 except;
          0.0.0.0/0;
        }
      }
      then {
        count reject_pref1_1;
        log;
        reject;
      }
    }
    term 2 {
      then {
        count reject_pref1_2;
        log;
        accept;
      }
    }
  }
}
```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
  unit 0 {
    family inet {
      filter {
        input fire1;
      }
      address 10.1.2.3/30;
    }
  }
}
```

3. If you are done configuring the device, commit your candidate configuration.

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

Verification

To confirm that the configuration is working properly, enter the **show firewall filter fire1** operational mode command. You can also display the log and individual counters separately by using the following forms of the command:

- **show firewall counter reject_pref1_1**
- **show firewall counter reject_pref1_2**
- **show firewall log**

Related Documentation

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Filter to Count IP Options Packets on page 111
- Example: Configuring a Filter to Count and Discard IP Options Packets on page 108

Example: Configuring a Filter to Count and Discard IP Options Packets

This example shows how to configure a standard stateless firewall to count packets.

- Requirements on page 108
- Overview on page 109
- Configuration on page 109
- Verification on page 111

Requirements

No special configuration beyond device initialization is required before configuring this example.

Because the filter term matches on *any* IP option value, the filter term can use the **count** nonterminating action without the **discard** terminating action or (alternatively) without requiring an interface on a 10-Gigabit Ethernet Modular Port Concentrator (MPC), 60-Gigabit Ethernet MPC, 60-Gigabit Queuing Ethernet MPC, or 60-Gigabit Ethernet Enhanced Queuing MPC on an MX Series router.

Overview

In this example, you use a standard stateless firewall filter to count and discard packets that include any IP option value but accept all other packets.

The IP option header field is an optional field in IPv4 headers only. The **ip-options** and **ip-options-except** match conditions are supported for standard stateless firewall filters and service filters only.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configure the Stateless Firewall Filter on page 109
- Apply the Stateless Firewall Filter to a Logical Interface on page 110
- Confirm and Commit Your Candidate Configuration on page 110

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter block_ip_options term 10 from ip-options any
set firewall family inet filter block_ip_options term 10 then count option_any
set firewall family inet filter block_ip_options term 10 then discard
set firewall family inet filter block_ip_options term 999 then accept
set interfaces ge-0/0/1 unit 0 family inet filter input block_ip_options
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.3/30
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter:

1. Create the stateless firewall filter **block_ip_options**.

```
[edit]
user@host# edit firewall family inet filter block_ip_options
```

2. Configure the first term to count and discard packets that include any IP options header fields.

```
[edit firewall family inet filter block_ip_options]
user@host# set term 10 from ip-options any
user@host# set term 10 then count option_any
user@host# set term 10 then discard
```

3. Configure the other term to accept all other packets.

```
[edit firewall family inet filter block_ip_options]
user@host# set term 999 then accept
```

Apply the Stateless Firewall Filter to a Logical Interface

Step-by-Step Procedure

To apply the stateless firewall filter to a logical interface:

1. Configure the logical interface to which you will apply the stateless firewall filter.

```
[edit]
user@host# edit interfaces ge-0/0/1 unit 0 family inet
```

2. Configure the interface address for the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set address 10.1.2.3/30
```

3. Apply the stateless firewall filter to the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set filter input block_ip_options
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter block_ip_options {
    term 10 {
      from {
        ip-options any;
      }
      then {
        count option_any;
        discard;
      }
    }
    term 999 {
      then accept;
    }
  }
}
```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
```



```
unit 0 {  
    family inet {  
        filter {  
            input block_ip_options;  
        }  
        address 10.1.2.3/30;  
    }  
}
```

3. If you are done configuring the device, commit your candidate configuration.

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

Verification

To confirm that the configuration is working properly, enter the **show firewall filter block_ip_options** operational mode command. To display the count of discarded packets separately, enter the **show firewall count option_any** form of the command.

Related Documentation

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Filter to Count Accepted and Rejected Packets on page 105
- Example: Configuring a Filter to Count IP Options Packets on page 111

Example: Configuring a Filter to Count IP Options Packets

This example shows how use a stateless firewall filter to count individual IP options packets:

- Requirements on page 111
- Overview on page 112
- Configuration on page 112
- Verification on page 116

Requirements

This example uses an interface on a 10-Gigabit Ethernet Modular Port Concentrator (MPC), 60-Gigabit Ethernet MPC, 60-Gigabit Queuing Ethernet MPC, or 60-Gigabit Ethernet Enhanced Queuing MPC on an MX Series router. This interface enables you to apply an IPv4 firewall filter (standard or service filter) that can use the **count**, **log**, and **syslog** nonterminating actions on packets that match a *specific ip-option* value without having to also use the **discard** terminating action.

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you use a stateless firewall filter to count IP options packets but not block any traffic. Also, the filter logs packets that have loose or strict source routing.

The IP option header field is an optional field in IPv4 headers only. The **ip-options** and **ip-options-except** match conditions are supported for standard stateless firewall filters and service filters only.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configure the Stateless Firewall Filter on page 113
- Apply the Stateless Firewall Filter to a Logical Interface on page 114
- Confirm and Commit Your Candidate Configuration on page 114

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter ip_options_filter term match_strict_source from ip-options
strict-source-route
set firewall family inet filter ip_options_filter term match_strict_source then count
strict_source_route
set firewall family inet filter ip_options_filter term match_strict_source then log
set firewall family inet filter ip_options_filter term match_strict_source then accept
set firewall family inet filter ip_options_filter term match_loose_source from ip-options
loose-source-route
set firewall family inet filter ip_options_filter term match_loose_source then count
loose_source_route
set firewall family inet filter ip_options_filter term match_loose_source then log
set firewall family inet filter ip_options_filter term match_loose_source then accept
set firewall family inet filter ip_options_filter term match_record from ip-options
record-route
set firewall family inet filter ip_options_filter term match_record then count record_route
set firewall family inet filter ip_options_filter term match_record then accept
set firewall family inet filter ip_options_filter term match_timestamp from ip-options
timestamp
set firewall family inet filter ip_options_filter term match_timestamp then count timestamp
set firewall family inet filter ip_options_filter term match_timestamp then accept
set firewall family inet filter ip_options_filter term match_router_alert from ip-options
router-alert
set firewall family inet filter ip_options_filter term match_router_alert then count
router_alert
set firewall family inet filter ip_options_filter term match_router_alert then accept
set firewall family inet filter ip_options_filter term match_all then accept
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.3/30
set interfaces ge-0/0/1 unit 0 family inet filter input ip_options_filter
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter `ip_option_filter`:

1. Create the stateless firewall filter `ip_option_filter`.


```
[edit]
user@host# edit firewall family inet filter ip_options_filter
```
2. Configure the first term to count, log, and accept packets with the `strict_source_route` IP optional header field.


```
[edit firewall family inet filter ip_option_filter]
user@host# set term match_strict_source from ip-options strict_source_route
user@host# set term match_strict_source then count strict_source_route
user@host# set term match_strict_source then log
user@host# set term match_strict_source then accept
```
3. Configure the next term to count, log, and accept packets with the `loose-source-route` IP optional header field.


```
[edit firewall family inet filter ip_option_filter]
user@host# set term match_loose_source from ip-options loose-source-route
user@host# set term match_loose_source then count loose_source_route
user@host# set term match_loose_source then log
user@host# set term match_loose_source then accept
```
4. Configure the next term to count and accept packets with the `record-route` IP optional header field.


```
[edit firewall family inet filter ip_option_filter]
user@host# set term match_record from ip-options record-route
user@host# set term match_record then count record_route
user@host# set term match_record then accept
```
5. Configure the next term to count and accept packets with the `timestamp` IP optional header field.


```
[edit firewall family inet filter ip_option_filter]
user@host# set term match_timestamp from ip-options timestamp
user@host# set term match_timestamp then count timestamp
user@host# set term match_timestamp then accept
```
6. Configure the next term to count and accept packets with the `router-alert` IP optional header field.


```
[edit firewall family inet filter ip_option_filter]
user@host# set term match_router_alert from ip-options router-alert
user@host# set term match_router_alert then count router_alert
user@host# set term match_router_alert then accept
```
7. Create the last term to accept any packet without incrementing any counters.


```
[edit firewall family inet filter ip_option_filter]
user@host# set term match_all then accept
```

Apply the Stateless Firewall Filter to a Logical Interface

Step-by-Step Procedure

To apply the stateless firewall filter to a logical interface:

1. Configure the logical interface to which you will apply the stateless firewall filter.

```
[edit]
user@host# edit interfaces ge-0/0/1 unit 0 family inet
```

2. Configure the interface address for the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set address 10.1.2.3/30
```

3. Apply the stateless firewall filter to the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set filter input ip_options_filter
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter ip_options_filter {
    term match_strict_source {
      from {
        ip-options strict-source-route;
      }
      then {
        count strict_source_route;
        log;
        accept;
      }
    }
    term match_loose_source {
      from {
        ip-options loose-source-route;
      }
      then {
        count loose_source_route;
        log;
        accept;
      }
    }
  }
  term match_record {
    from {
      ip-options record-route;
    }
    then {
      count record_route;
    }
  }
}
```

```

        accept;
    }
}
term match_timestamp {
    from {
        ip-options timestamp;
    }
    then {
        count timestamp;
        accept;
    }
}
term match_router_alert {
    from {
        ip-options router-alert;
    }
    then {
        count router_alert;
        accept;
    }
}
term match_all {
    then accept;
}
}
}

```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

[edit]
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            filter {
                input ip_option_filter;
            }
            address 10.1.2.3/30;
        }
    }
}

```

3. If you are done configuring the device, commit your candidate configuration.

```

[edit]
user@host# commit

```

Verification

To confirm that the configuration is working properly, enter the **show firewall filter ip_option_filter** operational mode command. You can also display the log and individual counters separately by using the following forms of the command:

- **show firewall counter strict_source_route**
- **show firewall counter loose_source_route**
- **show firewall counter record_route**
- **show firewall counter timestamp**
- **show firewall counter router_alert**
- **show firewall log**

Related Documentation

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Filter to Count Accepted and Rejected Packets on page 105
- Example: Configuring a Filter to Count and Discard IP Options Packets on page 108

CHAPTER 11

Standard Firewall Filters That Act on Packets

- Example: Configuring a Filter to Set the DSCP Bit to Zero on page 117
- Example: Configuring a Filter to Count and Sample Accepted Packets on page 120

Example: Configuring a Filter to Set the DSCP Bit to Zero

This example shows how to configure a standard stateless firewall filter based on the Differentiated Services code point (DSCP).

- Requirements on page 117
- Overview on page 117
- Configuration on page 117
- Verification on page 119

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you use a stateless firewall filter to match packets on DSCP bit patterns. If the DSCP is **2**, the packet is classified to the **best-effort** forwarding class, and the DSCP is set to **0**. If the DSCP is **3**, the packet is classified to the **best-effort** forwarding class.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configure the Stateless Firewall Filter on page 118
- Apply the Stateless Firewall Filter to a Logical Interface on page 118
- Confirm and Commit Your Candidate Configuration on page 119

CLI Quick Configuration To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall filter filter1 term 1 from dscp 2
set firewall filter filter1 term 1 then forwarding-class best-effort
set firewall filter filter1 term 1 then dscp 0
set firewall filter filter1 term 2 from dscp 3
set firewall filter filter1 term 2 then forwarding-class best-effort
set interfaces so-0/1/0 unit 0 family inet filter input filter1
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure To configure the stateless firewall filter **filter1**:

1. Create the stateless firewall filter.

```
[edit]
user@host# edit firewall filter filter1
```

2. Configure the first term to match a packet with a DSCP of **2**, change the DSCP to **0**, and classify the packet to the **best-effort** forwarding class.

```
[edit firewall filter filter1]
user@host# set term 1 from dscp 2
user@host# set term 1 then forwarding-class best-effort
user@host# set term 1 then dscp 0
```

3. Configure the other term to match a packet with a DSCP of **3** and classify the packet to the **best-effort** forwarding class.

```
[edit firewall filter filter1]
user@host# set term 2 from dscp 3
user@host# set term 2 then forwarding-class best-effort
```

Apply the Stateless Firewall Filter to a Logical Interface

Step-by-Step Procedure To apply the stateless firewall filter to the logical interface corresponding to the VPN routing and forwarding (VRF) instance:

1. Configure the logical interface to which you will apply the stateless firewall filter.

```
[edit]
user@host# edit interfaces so-0/1/0 unit 0 family inet
```

2. Apply the stateless firewall filter to the logical interface.

```
[ input filter1]
user@host# set filter input filter1
```


Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
filter filter1 {
  term term1 {
    from {
      dscp 2;
    }
    then {
      forwarding-class best-effort;
      dscp 0;
    }
  }
  term term2 {
    from {
      dscp 3;
    }
    then {
      forwarding-class best-effort;
    }
  }
}
```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
so-0/1/0 {
  unit 0 {
    family inet {
      filter input filter1;
    }
  }
}
```

3. If you are done configuring the device, commit your candidate configuration.

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

Verification

To confirm that the configuration is working properly, enter the following operational mode commands:

- **show class-of-service**—Displays the entire class-of-service (CoS) configuration, including system-chosen defaults.

- **show class-of-service classifier type dscp**—Displays only the classifiers of the DSCP for IPv4 type.

**Related
Documentation**

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Filter to Count and Sample Accepted Packets on page 120

Example: Configuring a Filter to Count and Sample Accepted Packets

This example shows how to configure a standard stateless firewall filter to count and sample accepted packets.

- Requirements on page 120
- Overview on page 120
- Configuration on page 120
- Verification on page 122

Requirements

No special configuration beyond device initialization is required before configuring this example.

Before you begin, configure traffic sampling by including the **sampling** statement at the **[edit forwarding-options]** hierarchy level.

Overview

In this example, you use a standard stateless firewall filter to count and sample all packets received on a logical interface.



NOTE: When you enable reverse path forwarding (RPF) on an interface with an input filter for firewall log and count, the input firewall filter does not log the packets rejected by RPF, although the rejected packets are counted. To log the rejected packets, use an RPF check fail filter.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configure the Stateless Firewall Filter on page 121
- Apply the Stateless Firewall Filter to a Logical Interface on page 121
- Confirm and Commit Your Candidate Configuration on page 122

CLI Quick Configuration To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter sam term all then count count_sam
set firewall family inet filter sam term all then sample
set interfaces at-2/0/0 unit 301 family inet address 10.1.2.3/30
set interfaces at-2/0/0 unit 301 family inet filter input sam
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure To configure the stateless firewall filter **sam**:

1. Create the stateless firewall filter **sam**.

[edit]
user@host# edit firewall family inet filter sam
2. Configure the term to count and sample all packets.

[edit firewall family inet filter sam]
user@host# set term all then count count_sam
user@host# set term all then sample

Apply the Stateless Firewall Filter to a Logical Interface

Step-by-Step Procedure To apply the stateless firewall filter to a logical interface:

1. Configure the logical interface to which you will apply the stateless firewall filter.

[edit]
user@host# edit interfaces ge-0/0/1 unit 0 family inet
2. Configure the interface address for the logical interface.

[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set address 10.1.2.3/30
3. Apply the stateless firewall filter to the logical interface.

[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set filter input sam



NOTE: The Junos OS does not sample packets originating from the router. If you configure a filter and apply it to the output side of an interface, then only the transit packets going through that interface are sampled. Packets that are sent from the Routing Engine to the Packet Forwarding Engine are not sampled.

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter sam {
    term all {
      then {
        count count_sam;
        sample; # default action is accept
      }
    }
  }
}
```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
interfaces {
  at-2/0/0 {
    unit 301 {
      family inet {
        filter {
          input sam;
        }
        address 10.1.2.3/30;
      }
    }
  }
}
```

3. If you are done configuring the device, commit your candidate configuration.

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

Verification

Confirm that the configuration is working properly.

- Displaying the Packet Counter on page 122
- Displaying the Firewall Filter Log Output on page 123
- Displaying the Sampling Output on page 123

Displaying the Packet Counter

Purpose Verify that the firewall filter is evaluating packets.

```

Action  user@host> show firewall filter sam
Filter:
Counters:
Name           Bytes           Packets
sam
sam-1          98              8028

```

Displaying the Firewall Filter Log Output

Purpose Display the packet header information for all packets evaluated by the firewall filter.

```

Action  user@host> show firewall log
Time     Filter  A Interface      Pro Source address  Destination address
23:09:09 -      A at-2/0/0.301    TCP 10.2.0.25      10.211.211.1:80
23:09:07 -      A at-2/0/0.301    TCP 10.2.0.25      10.211.211.1:56
23:09:07 -      A at-2/0/0.301    ICM 10.2.0.25      10.211.211.1:49552
23:02:27 -      A at-2/0/0.301    TCP 10.2.0.25      10.211.211.1:56
23:02:25 -      A at-2/0/0.301    TCP 10.2.0.25      10.211.211.1:80
23:01:22 -      A at-2/0/0.301    ICM 10.2.2.101     10.211.211.1:23251
23:01:21 -      A at-2/0/0.301    ICM 10.2.2.101     10.211.211.1:16557
23:01:20 -      A at-2/0/0.301    ICM 10.2.2.101     10.211.211.1:29471
23:01:19 -      A at-2/0/0.301    ICM 10.2.2.101     10.211.211.1:26873

```

Meaning This output file contains the following fields:

- **Time**—Time at which the packet was received (not shown in the default).
- **Filter**—Name of a filter that has been configured with the **filter** statement at the **[edit firewall]** hierarchy level. A hyphen (-) or the abbreviation **pfe** indicates that the packet was handled by the Packet Forwarding Engine. A space (no hyphen) indicates that the packet was handled by the Routing Engine.
- **A**—Filter action:
 - **A**—Accept (or next term)
 - **D**—Discard
 - **R**—Reject
- **Interface**—Interface on which the filter is configured.



NOTE: We strongly recommend that you always explicitly configure an action in the **then** statement.

- **Pro**—Packet's protocol name or number.
- **Source address**—Source IP address in the packet.
- **Destination address**—Destination IP address in the packet.

Displaying the Sampling Output

Purpose Verify that the sampling output contains appropriate data.

Action	File	Size	Last changed
	wtmp.0.gz	Size: 15017	Last changed: Dec 19 13:15:54
		Size: 493	Last changed: Nov 19 13:47:29
	wtmp.2.gz	Size: 57	Last changed: Oct 20 15:24:34
		Pipe through a command	

```
user@host> show log /var/tmp/sam
```

```
# Apr 7 15:48:50
```

Time	Dest addr	Src addr	Dest port	Src port	Proto	TOS	Pkt len	Intf num	IP frag	TCP flags
Apr 7 15:48:54	192.168.9.194	192.168.9.195	0	0	1	0x0	84	8	0x0	0x0
Apr 7 15:48:55	192.168.9.194	192.168.9.195	0	0	1	0x0	84	8	0x0	0x0
Apr 7 15:48:56	192.168.9.194	192.168.9.195	0	0	1	0x0	84	8	0x0	0x0

Related Documentation

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Filter to Set the DSCP Bit to Zero on page 117

CHAPTER 12

Standard Firewall Filters for Trusted Sources

- Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources on page 125
- Example: Configuring a Filter to Block Telnet and SSH Access on page 130
- Example: Configuring a Filter to Block TFTP Access on page 136
- Example: Configuring a Filter to Accept OSPF Packets from a Prefix on page 139
- Example: Configuring a Filter to Accept DHCP Packets Based on Address on page 141
- Example: Configuring a Filter to Block TCP Access to a Port Except From Specified BGP Peers on page 144

Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources

This example shows how to create a stateless firewall filter that protects the Routing Engine from traffic originating from untrusted sources.

- Requirements on page 125
- Overview on page 125
- Configuration on page 126
- Verification on page 128

Requirements

No special configuration beyond device initialization is required before configuring stateless firewall filters.

Overview

In this example, you create a stateless firewall filter called `protect-RE` that discards all traffic destined for the Routing Engine except SSH and BGP protocol packets from specified trusted sources. This example includes the following firewall filter terms:

- **ssh-term**—Accepts TCP packets with a source address of `192.168.122.0/24` and a destination port that specifies SSH.
- **bgp-term**—Accepts TCP packets with a source address of `10.2.1.0/24` and a destination port that specifies BGP.

- **discard-rest-term**—For all packets that are not accepted by **ssh-term** or **bgp-term**, creates a firewall filter log and system logging records, then discards all packets.



NOTE: You can move terms within the firewall filter using the `insert` command. See `insert` in the *Junos OS CLI User Guide*.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then paste them into the CLI at the **[edit]** hierarchy level.

```
[edit]
set firewall family inet filter protect-RE term ssh-term from source-address
  192.168.122.0/24
set firewall family inet filter protect-RE term ssh-term from protocol tcp
set firewall family inet filter protect-RE term ssh-term from destination-port ssh
set firewall family inet filter protect-RE term ssh-term then accept
set firewall family inet filter protect-RE term bgp-term from source-address 10.2.1.0/24
set firewall family inet filter protect-RE term bgp-term from protocol tcp
set firewall family inet filter protect-RE term bgp-term from destination-port bgp
set firewall family inet filter protect-RE term bgp-term then accept
set firewall family inet filter protect-RE term discard-rest-term then log
set firewall family inet filter protect-RE term discard-rest-term then syslog
set firewall family inet filter protect-RE term discard-rest-term then discard
set interfaces lo0 unit 0 family inet filter input protect-RE
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure the stateless firewall filter:

1. Create the stateless firewall filter.

```
[edit]
user@host# edit firewall family inet filter protect-RE
```

2. Create the first filter term.

```
[edit firewall family inet filter protect-RE]
user@host# edit term ssh-term
```

3. Define the protocol, destination port, and source address match conditions for the term.

```
[edit firewall family inet filter protect-RE term ssh-term]
user@host# set from protocol tcp destination-port ssh source-address
  192.168.122.0/24
```

4. Define the actions for the term.

```
[edit firewall family inet filter protect-RE term ssh-term]
user@host# set then accept
```


5. Create the second filter term.

```
[edit firewall family inet filter protect-RE]
user@host# edit term bgp-term
```

6. Define the protocol, destination port, and source address match conditions for the term.

```
[edit firewall family inet filter protect-RE term bgp-term]
user@host# set from protocol tcp destination-port bgp source-address 10.2.1.0/24
```

7. Define the action for the term.

```
[edit firewall family inet filter protect-RE term bgp-term]
user@host# set then accept
```

8. Create the third filter term.

```
[edit firewall family inet filter protect-RE]
user@host# edit term discard-rest-term
```

9. Define the action for the term.

```
[edit firewall family inet filter protect-RE term discard-rest]
user@host# set then log syslog discard
```

10. Apply the filter to the input side of the Routing Engine interface.

```
[edit]
user@host# set interfaces lo0 unit 0 family inet filter input protect-RE
```

Results Confirm your configuration by entering the **show firewall** command and the **show interfaces lo0** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show firewall
family inet {
  filter protect-RE {
    term ssh-term {
      from {
        source-address {
          192.168.122.0/24;
        }
        protocol tcp;
        destination-port ssh;
      }
      then accept;
    }
    term bgp-term {
      from {
        source-address {
          10.2.1.0/24;
        }
        protocol tcp;
        destination-port bgp;
      }
      then accept;
    }
  }
}
```

```
term discard-rest-term {
  then {
    log;
    syslog;
    discard;
  }
}

user@host# show interfaces lo0
unit 0 {
  family inet {
    filter {
      input protect-RE;
    }
    address 127.0.0.1/32;
  }
}
```

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

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

Verification

To confirm that the configuration is working properly, perform these tasks:

- Displaying Stateless Firewall Filter Configurations on page 128
- Verifying a Services, Protocols, and Trusted Sources Firewall Filter on page 128
- Displaying Stateless Firewall Filter Logs on page 129

Displaying Stateless Firewall Filter Configurations

Purpose	Verify the configuration of the firewall filter.
Action	From configuration mode, enter the show firewall command and the show interfaces lo0 command.
Meaning	Verify that the output shows the intended configuration of the firewall filter. In addition, verify that the terms are listed in the order in which you want the packets to be tested. You can move terms within a firewall filter by using the insert CLI command.

Verifying a Services, Protocols, and Trusted Sources Firewall Filter

Purpose	Verify that the actions of the firewall filter terms are taken.
Action	<p>Send packets to the device that match the terms. In addition, verify that the filter actions are <i>not</i> taken for packets that do not match.</p> <ul style="list-style-type: none">• Use the ssh <i>host-name</i> command from a host at an IP address that matches 192.168.122.0/24 to verify that you can log in to the device using only SSH from a host with this address prefix.

- Use the **show route summary** command to verify that the routing table on the device does not contain any entries with a protocol other than **Direct**, **Local**, **BGP**, or **Static**.

Sample Output

```
% ssh 192.168.249.71
%ssh host
user@host's password:
--- JUNOS 6.4-20040518.0 (JSERIES) #0: 2004-05-18 09:27:50 UTC

user@host>

user@host> show route summary
Router ID: 192.168.249.71

inet.0: 34 destinations, 34 routes (33 active, 0 holddown, 1 hidden)
      Direct:    10 routes,      9 active
        Local:     9 routes,      9 active
          BGP:    10 routes,    10 active
        Static:     5 routes,      5 active
...
```

Meaning Verify the following information:

- You can successfully log in to the device using SSH.
- The **show route summary** command does not display a protocol other than **Direct**, **Local**, **BGP**, or **Static**.

Displaying Stateless Firewall Filter Logs

Purpose Verify that packets are being logged. If you included the **log** or **syslog** action in a term, verify that packets matching the term are recorded in the firewall log or your system logging facility.

Action From operational mode, enter the **show firewall log** command.

Sample Output

```
user@host> show firewall log
Log :
Time      Filter  Action Interface  Protocol Src Addr      Dest Addr
15:11:02 pfe         D    ge-0/0/0.0   TCP      172.17.28.19  192.168.70.71
15:11:01 pfe         D    ge-0/0/0.0   TCP      172.17.28.19  192.168.70.71
15:11:01 pfe         D    ge-0/0/0.0   TCP      172.17.28.19  192.168.70.71
15:11:01 pfe         D    ge-0/0/0.0   TCP      172.17.28.19  192.168.70.71
...
```

Meaning Each record of the output contains information about the logged packet. Verify the following information:

- Under **Time**, the time of day the packet was filtered is shown.
- The **Filter** output is always **pfe**.
- Under **Action**, the configured action of the term matches the action taken on the packet—**A** (accept), **D** (discard), **R** (reject).

- Under **Interface**, the inbound (ingress) interface on which the packet arrived is appropriate for the filter.
- Under **Protocol**, the protocol in the IP header of the packet is appropriate for the filter.
- Under **Src Addr**, the source address in the IP header of the packet is appropriate for the filter.
- Under **Dest Addr**, the destination address in the IP header of the packet is appropriate for the filter.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- show route summary in the [Junos OS Routing Protocols and Policies Command Reference](#)
- show firewall in the [Junos OS Routing Protocols and Policies Command Reference](#)
- show firewall log in the [Junos OS Routing Protocols and Policies Command Reference](#)
- show interfaces (Loopback) in the [Junos OS Interfaces Command Reference](#)

Example: Configuring a Filter to Block Telnet and SSH Access

- Requirements on page 130
- Overview on page 130
- Configuration on page 130
- Verification on page 133

Requirements

You must have access to a remote host that has network connectivity with this router.

Overview

In this example, you create an IPv4 stateless firewall filter that logs and rejects Telnet or SSH access packets unless the packet is destined for or originates from the 192.168.1.0/24 subnet.

- To match packets destined for or originating from the **address 192.168.1.0/24** subnet, you use the **address 192.168.1.0/24** IPv4 match condition.
- To match packets destined for or originating from a TCP port, Telnet port, or SSH port, you use the **protocol tcp**, **port telnet**, and **telnet ssh** IPv4 match conditions.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configure the Stateless Firewall Filter on page 131
- Apply the Firewall Filter to the Loopback Interface on page 131
- Confirm and Commit Your Candidate Configuration on page 132

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter local_acl term terminal_access from address 192.168.1.0/24
set firewall family inet filter local_acl term terminal_access from protocol tcp
set firewall family inet filter local_acl term terminal_access from port ssh
set firewall family inet filter local_acl term terminal_access from port telnet
set firewall family inet filter local_acl term terminal_access then accept
set firewall family inet filter local_acl term terminal_access_denied from protocol tcp
set firewall family inet filter local_acl term terminal_access_denied from port ssh
set firewall family inet filter local_acl term terminal_access_denied from port telnet
set firewall family inet filter local_acl term terminal_access_denied then log
set firewall family inet filter local_acl term terminal_access_denied then reject
set firewall family inet filter local_acl term default-term then accept
set interfaces lo0 unit 0 family inet filter input local_acl
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter that selectively blocks Telnet and SSH access:

1. Create the stateless firewall filter **local_acl**.

```
[edit]
user@myhost# edit firewall family inet filter local_acl
```

2. Define the filter term **terminal_access**.

```
[edit firewall family inet filter local_acl]
user@myhost# set term terminal_access from address 192.168.1.0/24
user@myhost# set term terminal_access from protocol tcp
user@myhost# set term terminal_access from port ssh
user@myhost# set term terminal_access from port telnet
user@myhost# set term terminal_access then accept
```

3. Define the filter term **terminal_access_denied**.

```
[edit firewall family inet filter local_acl]
user@myhost# set term terminal_access_denied from protocol tcp
user@myhost# set term terminal_access_denied from port ssh
user@myhost# set term terminal_access_denied from port telnet
user@myhost# set term terminal_access_denied then log
user@myhost# set term terminal_access_denied then reject
user@myhost# set term default-term then accept
```

Apply the Firewall Filter to the Loopback Interface

Step-by-Step Procedure

- To apply the firewall filter to the loopback interface:

```
[edit]
```

```
user@myhost# set interfaces lo0 unit 0 family inet filter input local_acl
user@myhost# set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@myhost# show firewall
family inet {
  filter local_acl {
    term terminal_access {
      from {
        address {
          192.168.1.0/24;
        }
        protocol tcp;
        port [ssh telnet];
      }
      then accept;
    }
    term terminal_access_denied {
      from {
        protocol tcp;
        port [ssh telnet];
      }
      then {
        log;
        reject;
      }
    }
    term default-term {
      then accept;
    }
  }
}
```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@myhost# show interfaces
lo0 {
  unit 0 {
    family inet {
      filter {
        input local_acl;
      }
      address 127.0.0.1/32;
    }
  }
}
```

```
}
```

3. If you are done configuring the device, commit your candidate configuration.

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

Verification

Confirm that the configuration is working properly.

- Verifying Accepted Packets on page 133
- Verifying Logged and Rejected Packets on page 134

[Verifying Accepted Packets](#)

Purpose Verify that the actions of the firewall filter terms are taken.

- Action** 1. Clear the firewall log on your router.

```
user@myhost> clear firewall log
```

2. From a host at an IP address *within* the 192.168.1.0/24 subnet, use the **ssh hostname** command to verify that you can log in to the device using only SSH. This packet should be accepted, and the packet header information for this packet should not be logged in the firewall filter log buffer in the Packet Forwarding Engine.

```
user@host-A> ssh myhost

user@myhosts's password:
--- JUNOS 11.1-20101102.0 built 2010-11-02 04:48:46 UTC

% cli

user@myhost>
```

3. From a host at an IP address *within* the 192.168.1.0/24 subnet, use the **telnet hostname** command to verify that you can log in to your router using only Telnet. This packet should be accepted, and the packet header information for this packet should not be logged in the firewall filter log buffer in the Packet Forwarding Engine.

```
user@host-A> telnet myhost

Trying 192.168.249.71...
Connected to myhost-fxp0.acme.net.
Escape character is '^]'.

host (ttyp0)

login: user

Password:

--- JUNOS 11.1-20101102.0 built 2010-11-02 04:48:46 UTC

% cli

user@myhost>
```

4. Use the **show firewall log** command to verify that the routing table on the device does not contain any entries with a source address in the 192.168.1.0/24 subnet.

```
user@myhost> show firewall log
```

Verifying Logged and Rejected Packets

- Purpose** Verify that the actions of the firewall filter terms are taken.

- Action** 1. Clear the firewall log on your router.

```
user@myhost> clear firewall log
```

2. From a host at an IP address *outside of* the 192.168.1.0/24 subnet, use the **ssh *hostname*** command to verify that you cannot log in to the device using only SSH. This packet should be rejected, and the packet header information for this packet should be logged in the firewall filter log buffer in the Packet Forwarding Engine.

```
user@host-B ssh myhost
```

```
ssh: connect to host sugar port 22: Connection refused
--- JUNOS 11.1-20101102.0 built 2010-11-02 04:48:46 UTC
%
```

3. From a host at an IP address *outside of* the 192.168.1.0/24 subnet, use the **telnet *hostname*** command to verify that you can log in to the device using only Telnet. This packet should be rejected, and the packet header information for this packet should be logged in the firewall filter log buffer in the PFE.

```
user@host-B> telnet myhost
```

```
Trying 192.168.249.71...
telnet: connect to address 192.168.187.3: Connection refused
telnet: Unable to connect to remote host
%
```

4. Use the **show firewall log** command to verify that the routing table on the device does not contain any entries with a source address in the 192.168.1.0/24 subnet.

```
user@myhost> show firewall log
```

Time	Filter	Action	Interface	Protocol	Src Addr	Dest Addr
18:41:25	local_acl	R	fxp0.0	TCP	192.168.187.5	192.168.187.1
18:41:25	local_acl	R	fxp0.0	TCP	192.168.187.5	192.168.187.1
18:41:25	local_acl	R	fxp0.0	TCP	192.168.187.5	192.168.187.1
...						
18:43:06	local_acl	R	fxp0.0	TCP	192.168.187.5	192.168.187.1
18:43:06	local_acl	R	fxp0.0	TCP	192.168.187.5	192.168.187.1
18:43:06	local_acl	R	fxp0.0	TCP	192.168.187.5	192.168.187.1
...						

Related Documentation

- Logging of Packet Headers Evaluated by a Firewall Filter Term on page 58
- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources on page 125
- Example: Configuring a Filter to Block TFTP Access on page 136
- Example: Configuring a Filter to Accept OSPF Packets from a Prefix on page 139
- Example: Configuring a Filter to Accept DHCP Packets Based on Address on page 141

Example: Configuring a Filter to Block TFTP Access

- Requirements on page 136
- Overview on page 136
- Configuration on page 136
- Verification on page 138

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

By default, to decrease vulnerability to denial-of-service (DoS) attacks, the Junos OS filters and discards Dynamic Host Configuration Protocol (DHCP) or Bootstrap Protocol (BOOTP) packets that have a source address of 0.0.0.0 and a destination address of 255.255.255.255. This default filter is known as a unicast RPF check. However, some vendors' equipment automatically accepts these packets.

To interoperate with other vendors' equipment, you can configure a filter that checks for both of these addresses and overrides the default RPF-check filter by accepting these packets. In this example, you block Trivial File Transfer Protocol (TFTP) access, logging any attempts to establish TFTP connections.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see "Using the CLI Editor in Configuration Mode" on page 231.

To configure this example, perform the following tasks:

- Configure the Stateless Firewall Filter on page 137
- Apply the Firewall Filter to the Loopback Interface on page 137
- Confirm and Commit Your Candidate Configuration on page 137

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter tftp_access_control term one from protocol udp
set firewall family inet filter tftp_access_control term one from port tftp
set firewall family inet filter tftp_access_control term one then log
set firewall family inet filter tftp_access_control term one then discard
set interfaces lo0 unit 0 family inet filter input tftp_access_control
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter that selectively blocks TFTP access:

1. Create the stateless firewall filter **tftp_access_control**.

[edit]
user@host# **edit firewall family inet filter tftp_access_control**
2. Specify a match on packets received on UDP port 69.

[edit firewall family inet filter tftp_access_control]
user@host# **set term one from protocol udp**
user@host# **set term one from port tftp**
3. Specify that matched packets be logged to the buffer on the Packet Forwarding Engine and then discarded.

[edit firewall family inet filter tftp_access_control]
user@host# **set term one then log**
user@host# **set term one then discard**

Apply the Firewall Filter to the Loopback Interface

Step-by-Step Procedure

To apply the firewall filter to the loopback interface:

- [edit]
user@host# **set interfaces lo0 unit 0 family inet filter input tftp_access_control**
user@host# **set interfaces lo0 unit 0 family inet address 127.0.0.1/32**

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter tftp_access_control {
    term one {
      from {
        protocol udp;
        port tftp;
      }
      then {
        log;
        discard;
      }
    }
  }
}
```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
lo0 {
  unit 0 {
    family inet {
      filter {
        input tftp_access_control;
      }
      address 127.0.0.1/32;
    }
  }
}
```

3. If you are done configuring the device, commit your candidate configuration.

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

Verification

Confirm that the configuration is operating properly:

- Verifying Logged and Discarded Packets on page 138

Verifying Logged and Discarded Packets

Purpose Verify that the actions of the firewall filter terms are taken.

Action To

1. Clear the firewall log on your router.

```
user@myhost> clear firewall log
```

2. From another host, send a packet to UDP port 69 on this router.

**Related
Documentation**

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources on page 125
- Example: Configuring a Filter to Block Telnet and SSH Access on page 130
- Example: Configuring a Filter to Accept OSPF Packets from a Prefix on page 139
- Example: Configuring a Filter to Accept DHCP Packets Based on Address on page 141

Example: Configuring a Filter to Accept OSPF Packets from a Prefix

This example shows how to configure a standard stateless firewall filter to accept packets from a trusted source.

- Requirements on page 139
- Overview on page 139
- Configuration on page 139
- Verification on page 141

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you create a filter that accepts only OSPF packets from an address in the prefix 10.108.0.0/16, discarding all other packets with an **administratively-prohibited** ICMP message

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configure the Stateless Firewall Filter on page 139
- Apply the Firewall Filter to the Loopback Interface on page 140
- Confirm and Commit Your Candidate Configuration on page 140

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter ospf_filter term term1 from source-address 10.108.0.0/16
set firewall family inet filter ospf_filter term term1 from protocol ospf
set firewall family inet filter ospf_filter term term1 then accept
set firewall family inet filter ospf_filter term default-term then reject
  administratively-prohibited
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.3/30
set interfaces ge-0/0/1 unit 0 family inet filter input ospf_filter
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter **ospf_filter**:

1. Create the filter.

```
[edit]
user@host# edit firewall family inet filter ospf_filter
```

2. Configure the term that accepts packets.

```
[edit firewall family inet filter ospf_filter]
user@host# set term term1 from source-address 10.108.0.0/16
user@host# set term term1 from protocol ospf
user@host# set term term1 then accept
```

3. Configure the term that rejects all other packets.

```
[edit firewall family inet filter ospf_filter]
user@host# set term default_term then reject administratively-prohibited
```

Apply the Firewall Filter to the Loopback Interface

Step-by-Step Procedure

To apply the firewall filter to the loopback interface:

1. Configure the interface.

```
[edit]
user@host# edit interfaces ge-0/0/1 unit 0 family inet
```

2. Configure the logical interface IP address.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set address 10.1.2.3/30
```

3. Apply the filter to the input.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set filter input ospf_filter
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter ospf_filter {
    term term1 {
      from {
        source-address {
          10.108.0.0/16;
        }
        protocol ospf;
      }
      then {
        accept;
      }
    }
  }
  term default_term {
    then {
      reject administratively-prohibited; # default reject action
    }
  }
}
```

```

    }
  }
}

```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

[edit]
user@host# show interfaces
lo0 {
  unit 0 {
    family inet {
      filter {
        input ospf_filter;
      }
      address 10.1.2.3/30;
    }
  }
}

```

3. If you are done configuring the device, commit your candidate configuration.

```

[edit]
user@host# commit

```

Verification

To confirm that the configuration is working properly, enter the **show firewall filter ospf_filter** operational mode command.

Related Documentation

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources on page 125
- Example: Configuring a Filter to Block Telnet and SSH Access on page 130
- Example: Configuring a Filter to Block TFTP Access on page 136
- Example: Configuring a Filter to Accept DHCP Packets Based on Address on page 141

Example: Configuring a Filter to Accept DHCP Packets Based on Address

This example shows how to configure a standard stateless firewall filter to accept packets from a trusted source.

- Requirements on page 142
- Overview on page 142
- Configuration on page 142
- Verification on page 144

Requirements

This example is supported on MX Series routers only.

Overview

In this example, you create a filter (**rpf_dhcp**) that accepts DHCP packets with a source address of **0.0.0.0** and a destination address of **255.255.255.255**.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see "Using the CLI Editor in Configuration Mode" on page 231.

- Configure the Stateless Firewall Filter on page 142
- Apply the Firewall Filter to the Loopback Interface on page 143
- Confirm and Commit Your Candidate Configuration on page 143

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter rpf_dhcp term dhcp_term from source-address 0.0.0.0/32
set firewall family inet filter rpf_dhcp term dhcp_term from destination-address
  255.255.255.255/32
set firewall family inet filter rpf_dhcp term dhcp_term then accept
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.3/30
set interfaces ge-0/0/1 unit 0 family inet filter input sam
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter:

1. Create the stateless firewall filter **rpf_dhcp**.

[edit]
user@host# **edit firewall family inet filter rpf_dhcp**
2. Configure the term to match packets with a source address of **0.0.0.0** and a destination address of **255.255.255.255**.

[edit firewall family inet filter rpf_dhcp]
user@host# **set term dhcp_term from source-address 0.0.0.0/32**
user@host# **set term dhcp_term from destination-address 255.255.255.255/32**
3. Configure the term to accept packets that match the specified conditions.

[edit firewall family inet filter rpf_dhcp]
set term dhcp_term then accept

Apply the Firewall Filter to the Loopback Interface

Step-by-Step Procedure

To apply the filter to the input at the loopback interface:

1. Apply the **rpf_dhcp** filter if packets are not arriving on an expected path.

```
[edit]
user@host# set interfaces lo0 unit 0 family inet rpf-check fail-filter rpf_dhcp
```
2. Configure an address for the loopback interface.

```
[edit]
user@host# set interfaces lo0 unit 0 family inet address 127.0.0.1/32
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter rpf_dhcp {
    term dhcp_term {
      from {
        source-address {
          0.0.0.0/32;
        }
        destination-address {
          255.255.255.255/32;
        }
      }
      then accept;
    }
  }
}
```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
lo0 {
  unit 0 {
    family inet {
      filter {
        rpf-check {
          fail-filter rpf_dhcp;
          mode loose;
        }
      }
      address 127.0.0.1/32;
    }
  }
}
```

```
}  
}
```

3. When you are done configuring the device, commit your candidate configuration.

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

Verification

To confirm that the configuration is working properly, enter the **show firewall** operational mode command.

Related Documentation

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources on page 125
- Example: Configuring a Filter to Block Telnet and SSH Access on page 130
- Example: Configuring a Filter to Block TFTP Access on page 136
- Example: Configuring a Filter to Accept OSPF Packets from a Prefix on page 139

Example: Configuring a Filter to Block TCP Access to a Port Except From Specified BGP Peers

This example shows how to configure a standard stateless firewall filter that limits certain TCP and ICMP traffic destined for the Routing Engine.

- Requirements on page 144
- Overview on page 144
- Configuration on page 145
- Verification on page 148

Requirements

No special configuration beyond device initialization is required before configuring this example.

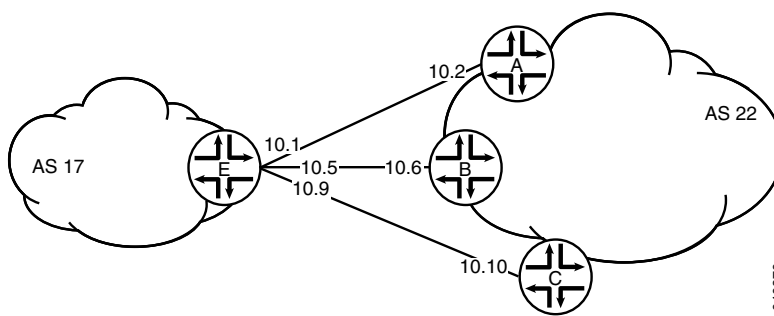
Overview

In this example, you create a stateless firewall filter that blocks all TCP connection attempts to port **179** from all requesters except the specified BGP peers.

The stateless firewall filter **filter_bgp179** matches all packets from the directly connected interfaces on Device A and Device B to the destination port number 179.

Figure 2 on page 145 shows the topology used in this example. Device C attempts to make a TCP connection to Device E. Device E blocks the connection attempt. This example shows the configuration on Device C and Device E.

Figure 2: Typical Network with BGP Peer Sessions



Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configuring the Stateless Firewall Filter on page 146
- Applying the Firewall Filter to the Loopback Interface on page 146
- Confirming and Committing Your Candidate Configuration on page 146

CLI Quick Configuration To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the [edit] hierarchy level.

```

Device C
set interfaces ge-1/2/0 unit 10 description to-E
set interfaces ge-1/2/0 unit 10 family inet address 10.10.10.10/30
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 17
set protocols bgp group external-peers neighbor 10.10.10.9
set routing-options autonomous-system 22

Device E
set interfaces ge-1/2/0 unit 0 description to-A
set interfaces ge-1/2/0 unit 0 family inet address 10.10.10.1/30
set interfaces ge-1/2/1 unit 5 description to-B
set interfaces ge-1/2/1 unit 5 family inet address 10.10.10.5/30
set interfaces ge-1/0/0 unit 9 description to-C
set interfaces ge-1/0/0 unit 9 family inet address 10.10.10.9/30
set interfaces lo0 unit 2 family inet filter input filter_bgp179
set interfaces lo0 unit 2 family inet address 192.168.0.1/32
set protocols bgp group external-peers type external
set protocols bgp group external-peers peer-as 22
set protocols bgp group external-peers neighbor 10.10.10.2
set protocols bgp group external-peers neighbor 10.10.10.6
set protocols bgp group external-peers neighbor 10.10.10.10
set routing-options autonomous-system 17
set firewall family inet filter filter_bgp179 term 1 from source-address 10.10.10.2/32
set firewall family inet filter filter_bgp179 term 1 from source-address 10.10.10.6/32
set firewall family inet filter filter_bgp179 term 1 from destination-port bgp
set firewall family inet filter filter_bgp179 term 1 then accept

```

```
set firewall family inet filter filter_bgp179 term 2 then reject
```

Configuring the Stateless Firewall Filter

Step-by-Step Procedure To configure the stateless firewall filter that blocks all TCP connection attempts to port 179 from all requestors except specified BGP peers:

1. Create the stateless firewall filter **filter_bgp179**.

```
[edit]  
user@E# edit firewall family inet filter filter_bgp179
```
2. Define the filter term that accepts TCP connection attempts to port 179 from the specified BGP peers.

```
[edit firewall family inet filter filter_bgp179]  
user@E# set term 1 from source-address 10.10.10.2/32  
user@E# set term 1 from source-address 10.10.10.6/32  
user@E# set term 1 from destination-port bgp  
user@E# set term 1 then accept
```
3. Define the other filter term to reject packets from other sources.

```
[edit firewall family inet filter filter_bgp179]  
user@E# set term 2 then reject
```

Applying the Firewall Filter to the Loopback Interface

- Step-by-Step Procedure**
- To apply the firewall filter to the loopback interface:

```
[edit]  
user@E# set interfaces lo0 unit 2 family inet filter input filter_bgp179  
user@E# set interfaces lo0 unit 2 family inet address 192.168.0.1/32
```

Confirming and Committing Your Candidate Configuration

Step-by-Step Procedure To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]  
user@E# show firewall  
family inet {  
  filter filter_bgp179 {  
    term 1 {  
      from {  
        source-address {  
          10.10.10.2/32;  
          10.10.10.6/32;  
        }  
        destination-port bgp;  
      }  
      then accept;  
    }  
    term 2 {
```

```

        then {
            reject;
        }
    }
}

```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

[edit]
user@E# show interfaces
lo0 {
    unit 2 {
        family inet {
            filter {
                input filter_bgp179;
            }
            address 192.168.0.1/32;
        }
    }
}
ge-1/2/0 {
    unit 0 {
        description to-A;
        family inet {
            address 10.10.10.1/30;
        }
    }
}
ge-1/2/1 {
    unit 5 {
        description to-B;
        family inet {
            address 10.10.10.5/30;
        }
    }
}
ge-1/0/0 {
    unit 9 {
        description to-C;
        family inet {
            address 10.10.10.9/30;
        }
    }
}
}

```

3. Confirm the configuration of the interface by entering the **show protocols** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

[edit]
user@E# show protocols
bgp {
    group external-peers {
        type external;
    }
}

```

```
peer-as 22;
neighbor 10.10.10.2;
neighbor 10.10.10.6;
neighbor 10.10.10.10;
}
}
```

4. When you are done configuring the device, commit your candidate configuration.

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

Verification

Confirm that the configuration is working properly.

- Verifying That the Filter Is Configured on page 148
- Verifying the TCP Connections on page 148
- Monitoring Traffic on the Interfaces on page 148

Verifying That the Filter Is Configured

Purpose Verify that the configuration is working properly.

Action From operational mode, run the **show firewall filter** command.

```
user@E> show firewall filter filter_bgp179
Filter: filter_bgp179
```

Verifying the TCP Connections

Purpose Verify the TCP connections.

Action From operational mode, run the **show system connections extensive** command on Device C and Device E.

The output on Device C shows the attempt to establish a TCP connection. The output on Device E shows that connections are established with Devices A and B only.

```
user@C> show system connections extensive | match 10.10.10
```

tcp4	0	0	10.10.10.9.51872	10.10.10.10.179	SYN_SENT
------	---	---	------------------	-----------------	----------

```
user@E> show system connections extensive | match 10.10.10
```

tcp4	0	0	10.10.10.5.179	10.10.10.6.62096	ESTABLISHED
tcp4	0	0	10.10.10.6.62096	10.10.10.5.179	ESTABLISHED
tcp4	0	0	10.10.10.1.179	10.10.10.2.61506	ESTABLISHED
tcp4	0	0	10.10.10.2.61506	10.10.10.1.179	ESTABLISHED

Monitoring Traffic on the Interfaces

Purpose Compare the traffic on an interface that establishes a TCP connection with the traffic on an interface that does not establish a TCP connection.

Action From operational mode, run the **monitor traffic** command on Device E's interface to Device B and on Device E's interface to Device C. In the first example, acknowledgment (**ack**) messages are received. In the second example, **ack** messages are not received.

```
user@E> monitor traffic size 1500 interface ge-1/2/1.5
19:02:49.700912 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: P
3330573561:3330573580(19) ack 915601686 win 16384 <nop,nop,timestamp 1869518816
1869504850>: BGP, length: 19
19:02:49.801244 In IP 10.10.10.6.62096 > 10.10.10.5.bgp: . ack 19 win 16384
<nop,nop,timestamp 1869518916 1869518816>
19:03:03.323018 In IP 10.10.10.6.62096 > 10.10.10.5.bgp: P 1:20(19) ack 19 win
16384 <nop,nop,timestamp 1869532439 1869518816>: BGP, length: 19
19:03:03.422418 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: . ack 20 win 16384
<nop,nop,timestamp 1869532539 1869532439>
19:03:17.220162 Out IP 10.10.10.5.bgp > 10.10.10.6.62096: P 19:38(19) ack 20 win
16384 <nop,nop,timestamp 1869546338 1869532439>: BGP, length: 19
19:03:17.320501 In IP 10.10.10.6.62096 > 10.10.10.5.bgp: . ack 38 win 16384
<nop,nop,timestamp 1869546438 1869546338>
```

```
user@E> monitor traffic size 1500 interface ge-1/0/0.9
18:54:20.175471 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869009240 0,sackOK,eol>
18:54:23.174422 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869012240 0,sackOK,eol>
18:54:26.374118 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,nop,wscale 0,nop,nop,timestamp 1869015440 0,sackOK,eol>
18:54:29.573799 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,sackOK,eol>
18:54:32.773493 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,sackOK,eol>
18:54:35.973185 Out IP 10.10.10.9.61335 > 10.10.10.10.bgp: S 573929123:573929123(0)
win 16384 <mss 1460,sackOK,eol>
```

- Related Documentation**
- Basic Uses for Standard Firewall Filters on page 24
 - Example: Configuring a Stateless Firewall Filter to Protect Against TCP and ICMP Floods on page 151
 - Example: Configuring a Filter to Accept Packets Based on IPv6 TCP Flags on page 158

CHAPTER 13

Standard Firewall Filters for Flood Prevention

- Example: Configuring a Stateless Firewall Filter to Protect Against TCP and ICMP Floods on page 151
- Example: Configuring a Filter to Accept Packets Based on IPv6 TCP Flags on page 158

Example: Configuring a Stateless Firewall Filter to Protect Against TCP and ICMP Floods

This example shows how to create a stateless firewall filter that protects against TCP and ICMP denial-of-service attacks.

- Requirements on page 151
- Overview on page 151
- Configuration on page 152
- Verification on page 155

Requirements

No special configuration beyond device initialization is required before configuring stateless firewall filters.

Overview

In this example, you create a stateless firewall filter called **protect-RE** that polices TCP and ICMP packets. This example includes the following policers:

- **tcp-connection-policer**—Limits the traffic rate of the TCP packets to 500,000 bps and the burst size to 15,000 bytes. Packets that exceed the traffic rate are discarded.
- **icmp-policer**—Limits the traffic rate of the ICMP packets to 1,000,000 bps and the burst size to 15,000 bytes. Packets that exceed the traffic rate are discarded.

When specifying limits, the bandwidth limit can be from 32,000 bps to 32,000,000,000 bps and the burst-size limit can be from 1,500 bytes through 100,000,000 bytes. Use the following abbreviations when specifying limits: k (1,000), m (1,000,000), and g (1,000,000,000).

Each policer is incorporated into the action of a filter term. This example includes the following terms:

- **tcp-connection-term**—Policies certain TCP packets with a source address of 192.168.122.0/24 or 10.2.1.0/24. These addresses are defined in the **trusted-addresses** prefix list.

Policed packets include connection request packets (SYN and ACK flag bits equal 1 and 0), connection release packets (FIN flag bit equals 1), and connection reset packets (RST flag bit equals 1).

- **icmp-term**—Policies echo request packets, echo response packets, unreachable packets, and time-exceeded packets. All of these ICMP packets are counted in the **icmp-counter** counter.



NOTE: You can move terms within the firewall filter by using the **insert** command. See **insert** in the *Junos OS CLI User Guide*.

If you want to include the terms created in this procedure in the **protect-RE** firewall filter configured in “Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources” on page 125, perform the configuration tasks in this example first. Then configure the terms as described in “Example: Configuring a Stateless Firewall Filter to Accept Traffic from Trusted Sources” on page 125. This approach ensures that the rate-limiting terms are included as the first two terms in the firewall filter.



NOTE: You can move terms within the firewall filter by using the **insert** command. See **insert** in the *Junos OS CLI User Guide*.

Configuration

CLI Quick Configuration

To quickly configure the stateless firewall filter, copy the following commands to a text file, remove any line breaks, and then paste the commands into the CLI.

```
[edit]
set firewall family inet filter protect-RE term tcp-connection-term from source-prefix-list
trusted-addresses
set firewall family inet filter protect-RE term tcp-connection-term from protocol tcp
set firewall family inet filter protect-RE term tcp-connection-term from tcp-flags "(syn
& !ack) | fin | rst"
set firewall family inet filter protect-RE term tcp-connection-term then policer
tcp-connection-policer
set firewall family inet filter protect-RE term tcp-connection-term then accept
set firewall family inet filter protect-RE term icmp-term from protocol icmp
set firewall family inet filter protect-RE term icmp-term from icmp-type echo-request
set firewall family inet filter protect-RE term icmp-term from icmp-type echo-reply
set firewall family inet filter protect-RE term icmp-term from icmp-type unreachable
set firewall family inet filter protect-RE term icmp-term from icmp-type time-exceeded
set firewall family inet filter protect-RE term icmp-term then policer icmp-policer
set firewall family inet filter protect-RE term icmp-term then count icmp-counter
set firewall family inet filter protect-RE term icmp-term then accept
```

```

set firewall policer tcp-connection-policer filter-specific
set firewall policer tcp-connection-policer if-exceeding bandwidth-limit 1m
set firewall policer tcp-connection-policer if-exceeding burst-size-limit 15k
set firewall policer tcp-connection-policer then discard
set firewall policer icmp-policer filter-specific
set firewall policer icmp-policer if-exceeding bandwidth-limit 1m
set firewall policer icmp-policer if-exceeding burst-size-limit 15k
set firewall policer icmp-policer then discard
set policy-options prefix-list trusted-addresses 10.2.1.0/24
set policy-options prefix-list trusted-addresses 192.168.122.0/24

```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure stateless firewall filter policers:

1. Define the first policer.

```

[edit]
user@host# edit firewall policer tcp-connection-policer

```
2. Define the action for the policer.

```

[edit firewall policer tcp-connection-policer]
user@host# set then discard

```
3. Define the rate limits for the policer.

```

[edit firewall policer tcp-connection-policer]
user@host# set filter-specific
user@host# set if-exceeding burst-size-limit 15k bandwidth-limit 1m

```
4. Define the second policer.

```

[edit]
user@host# edit firewall policer icmp-policer

```
5. Define the action for the policer.

```

[edit firewall policer icmp-policer]
user@host# set then discard

```
6. Set the rate limits for the policer.

```

[edit firewall policer icmp-policer]
user@host# set filter-specific
user@host# set if-exceeding burst-size-limit 15k bandwidth-limit 1m

```
7. Define the prefix list.

```

[edit]
user@host# set policy-options prefix-list trusted-addresses 192.168.122.0/24
user@host# set policy-options prefix-list trusted-addresses 10.2.1.0/24

```
8. Create the stateless firewall filter.

```

[edit]
user@host# edit firewall family inet filter protect-RE

```

9. Define the first term for the filter.

```
[edit firewall family inet filter protect-RE]
user@host# edit term tcp-connection-term
```

10. Define the source address match condition for the term.

```
[edit firewall family inet filter protect-RE term tcp-connection-term]
user@host# set from source-prefix-list trusted-addresses
```

11. Define protocol match conditions for the term.

```
[edit firewall family inet filter protect-RE term tcp-connection-term]
user@host# set from protocol tcp tcp-flags "(syn & !ack) | fin | rst"
```

12. Define the actions for the term.

```
[edit firewall family inet filter protect-RE term tcp-connection-term]
user@host# set then policer tcp-connection-policer accept
```

13. Define the second term.

```
[edit]
user@host# edit firewall family inet filter protect-RE term icmp-term
```

14. Define the protocol for the term.

```
[edit firewall family inet filter protect-RE term icmp-term]
user@host# set from protocol icmp
```

15. Define the match conditions for the term.

```
[edit firewall family inet filter protect-RE term icmp-term]
user@host# set from icmp-type [echo-request echo-reply unreachable
time-exceeded]
```

16. Define the action for the term.

```
[edit firewall family inet filter protect-RE term icmp-term]
user@host# set then policer icmp-policer count icmp-counter accept
```

Results Confirm your configuration by entering the **show firewall** command and the **show policy-options** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show firewall
family inet {
  filter protect-RE {
    term tcp-connection-term {
      from {
        source-prefix-list {
          trusted-addresses;
        }
        protocol tcp;
        tcp-flags "(syn & !ack) | fin | rst";
      }
      then {
        policer tcp-connection-policer;
        accept;
      }
    }
  }
}
```

```

    }
  }
  term icmp-term {
    from {
      protocol icmp;
      icmp-type [ echo-request echo-reply unreachable time-exceeded ];
    }
    then {
      policer icmp-policer;
      count icmp-counter;
      accept;
    }
  }
}
policer tcp-connection-policer {
  filter-specific;
  if-exceeding {
    bandwidth-limit 1m;
    burst-size-limit 15k;
  }
  then discard;
}
policer icmp-policer {
  filter-specific;
  if-exceeding {
    bandwidth-limit 1m;
    burst-size-limit 15k;
  }
  then discard;
}

user@host# show policy-options
prefix-list trusted-addresses {
  10.2.1.0/24;
  192.168.122.0/24;
}

```

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

Verification

Confirm that the configuration is working properly.

- Displaying Stateless Firewall Filter Configurations on page 155
- Verifying a TCP and ICMP Flood Firewall Filter on page 156
- Displaying Firewall Filter Statistics on page 157

Displaying Stateless Firewall Filter Configurations

Purpose Verify the configuration of the firewall filter.

Action From configuration mode, enter the **show firewall** command.

Meaning Verify that the output shows the intended configuration of the firewall filter. In addition, verify that the terms are listed in the order in which you want the packets to be tested. You can move terms within a firewall filter by using the **insert** CLI command.

Verifying a TCP and ICMP Flood Firewall Filter

Purpose Verify that the actions of the firewall filter terms are taken.

Action Send packets to the device that match the terms. In addition, verify that the filter actions are *not* taken for packets that do not match.

- Verify that the device can establish only TCP sessions with a host at an IP address that matches 192.168.122.0/24 or 10.2.1.0/24. For example, log in to the device with the **telnet *host-name*** command from another host with one of these address prefixes.
- Use the **ping *host-name*** command to verify that the device responds only to ICMP packets (such as ping requests) that do not exceed the policer traffic rates.
- Use the **ping *host-name* size *bytes*** command to exceed the policer traffic rates by sending ping requests with large data payloads.

Sample Output

```
user@host> telnet 192.168.249.71
Trying 192.168.249.71...
Connected to host.acme.net.
Escape character is '^]'.

host (ttyp0)

login: user
Password:

--- JUNOS 6.4-20040521.1 built 2004-05-21 09:38:12 UTC

user@host>

user@host> ping 192.168.249.71
PING host-ge-000.acme.net (192.168.249.71): 56 data bytes
64 bytes from 192.168.249.71: icmp_seq=0 ttl=253 time=11.946 ms
64 bytes from 192.168.249.71: icmp_seq=1 ttl=253 time=19.474 ms
64 bytes from 192.168.249.71: icmp_seq=2 ttl=253 time=14.639 ms
...
```

```
user@host> ping 192.168.249.71 size 20000
PING host-ge-000.acme.net (192.168.249.71): 20000 data bytes
^C
--- host-ge-000.acme.net ping statistics ---
12 packets transmitted, 0 packets received, 100% packet loss
```

Meaning Verify the following information:

- You can successfully log in to the device using Telnet.
- The device sends responses to the **ping host** command.
- The device does not send responses to the **ping host size 20000** command.

Displaying Firewall Filter Statistics

Purpose Verify that packets are being policed and counted.

Action From operational mode, enter the **show firewall filter *filter-name*** command.

Sample Output

```
user@host> show firewall filter protect-RE
Filter: protect-RE
Counters:
Name                               Bytes          Packets
icmp-counter                        1040000         5600
Policers:
Name                               Packets
tcp-connection-policer            643254873
icmp-policer                       7391
```

Meaning Verify the following information:

- Next to **Filter**, the name of the firewall filter is correct.
- Under **Counters**:
 - Under **Name**, the names of any counters configured in the firewall filter are correct.
 - Under **Bytes**, the number of bytes that match the filter term containing the **count *counter-name*** action are shown.
 - Under **Packets**, the number of packets that match the filter term containing the **count *counter-name*** action are shown.
- Under **Policers**:
 - Under **Name**, the names of any policers configured in the firewall filter are correct.
 - Under **Packets**, the number of packets that match the conditions specified for the policer are shown.

- Related Documentation**
- “Two-Color Policer Configuration Overview” in the [Junos OS Firewall Filter and Policer Configuration Guide](#).
 - [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
 - show firewall in the [Junos OS Routing Protocols and Policies Command Reference](#)
 - ping in the [Junos OS System Basics and Services Command Reference](#).
 - telnet in the [Junos OS System Basics and Services Command Reference](#).

Example: Configuring a Filter to Accept Packets Based on IPv6 TCP Flags

This example shows how to configure a standard stateless firewall filter to accept packets from a trusted source.

- Requirements on page 158
- Overview on page 158
- Configuration on page 158
- Verification on page 160

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you create a filter that accepts packets with specific IPv6 TCP flags.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see "Using the CLI Editor in Configuration Mode" on page 231.

- Configure the Stateless Firewall Filter on page 158
- Apply the Firewall Filter to the Loopback Interface on page 159
- Confirm and Commit Your Candidate Configuration on page 159

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet6 filter tcp_filter term 1 from next-header tcp
set firewall family inet6 filter tcp_filter term 1 from tcp-flags syn
set firewall family inet6 filter tcp_filter term 1 then count tcp_syn_pkt
set firewall family inet6 filter tcp_filter term 1 then log
set firewall family inet6 filter tcp_filter term 1 then accept
set interfaces lo0 unit 0 family inet6 filter input tcp_filter
set interfaces lo0 unit 0 family inet6 address ::10.34.1.0/120
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the firewall filter

1. Create the IPv6 stateless firewall filter **tcp_filter**.

```
[edit]
user@host# edit firewall family inet6 filter tcp_filter
```

2. Specify that a packet matches if it is the initial packet in a TCP session and the next header after the IPv6 header is type TCP.

```
[edit firewall family inet6 filter tcp_filter]
user@host# set term 1 from next-header tcp
```



```
user@host# set term 1 from tcp-flags syn
```

3. Specify that matched packets are counted, logged to the buffer on the Packet Forwarding Engine, and accepted.

```
[edit firewall family inet6 filter tcp_filter]
user@host# set term 1 then count tcp_syn_pkt
user@host# set term 1 then log
user@host# set term 1 then accept
```

Apply the Firewall Filter to the Loopback Interface

Step-by-Step Procedure

To apply the firewall filter to the loopback interface:

- [edit]
user@host# set interfaces lo0 unit 0 family inet6 filter input tcp_filter
user@host# set interfaces lo0 unit 0 family inet6 address ::10.34.1.0/120

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet6 {
  filter tcp_filter {
    term 1 {
      from {
        next-header tcp;
        tcp-flags syn;
      }
      then {
        count tcp_syn_pkt;
        log;
        accept;
      }
    }
  }
}
```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
lo0 {
  unit 0 {
    family inet6 {
      filter {
        input tcp_filter;
      }
      address ::10.34.1.0/120;
    }
  }
}
```

```
    }  
  }  
}
```

3. When you are done configuring the device, commit your candidate configuration.

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

Verification

To confirm that the configuration is working properly, enter the **show firewall** operational mode command.

Related Documentation

- Basic Uses for Standard Firewall Filters on page 24
- Example: Configuring a Stateless Firewall Filter to Protect Against TCP and ICMP Floods on page 151
- Example: Configuring a Filter to Block TCP Access to a Port Except From Specified BGP Peers on page 144

Standard Firewall Filters for Fragment Handling

- Example: Configuring a Stateless Firewall Filter to Handle Fragments on page 161

Example: Configuring a Stateless Firewall Filter to Handle Fragments

This example shows how to create a stateless firewall filter that handles packet fragments.

- Requirements on page 161
- Overview on page 161
- Configuration on page 162
- Verification on page 165

Requirements

No special configuration beyond device initialization is required before configuring stateless firewall filters.

Overview

In this example, you create a stateless firewall filter called **fragment-RE** that accepts fragmented packets originating from 10.2.1.0/24 and destined for the BGP port. This example includes the following firewall filter terms:

- **not-from-prefix-term**—Discards packets that are not from 10.2.1.0/24 to ensure that subsequent terms in the firewall filter are matched against packets from 10.2.1.0/24 only.
- **small-offset-term**—Discards small (1–5) offset packets to ensure that subsequent terms in the firewall filter can be matched against all the headers in the packet. In addition, the term adds a record to the system logging destinations for the firewall facility.
- **not-fragmented-term**—Accepts unfragmented TCP packets with a destination port that specifies the BGP protocol. A packet is considered unfragmented if the MF flag is not set and the fragment offset equals 0.

- **first-fragment-term**—Accepts the first fragment of a fragmented TCP packet with a destination port that specifies the BGP protocol.
- **fragment-term**—Accepts all fragments that were not discarded by **small-offset-term**. (packet fragments 6–8191). However, only those fragments that are part of a packet containing a first fragment accepted by **first-fragment-term** are reassembled by the destination device.

Packet fragments offset can be from 1 through 8191.



NOTE: You can move terms within the firewall filter by using the **insert** command. For more information, see “insert” in the *Junos OS CLI User Guide*.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **edit** hierarchy level.

```
[edit]
set firewall family inet filter fragment-RE term not-from-prefix-term from source-address 0.0.0.0/0
set firewall family inet filter fragment-RE term not-from-prefix-term from source-address 10.2.1.0/24 except
set firewall family inet filter fragment-RE term not-from-prefix-term then discard
set firewall family inet filter fragment-RE term small-offset-term from fragment-offset 1-5
set firewall family inet filter fragment-RE term small-offset-term then syslog
set firewall family inet filter fragment-RE term small-offset-term then discard
set firewall family inet filter fragment-RE term not-fragmented-term from fragment-offset 0
set firewall family inet filter fragment-RE term not-fragmented-term from fragment-flags "!more-fragments"
set firewall family inet filter fragment-RE term not-fragmented-term from protocol tcp
set firewall family inet filter fragment-RE term not-fragmented-term from destination-port bgp
set firewall family inet filter fragment-RE term not-fragmented-term then accept
set firewall family inet filter fragment-RE term first-fragment-term from first-fragment
set firewall family inet filter fragment-RE term first-fragment-term from protocol tcp
set firewall family inet filter fragment-RE term first-fragment-term from destination-port bgp
set firewall family inet filter fragment-RE term first-fragment-term then accept
set firewall family inet filter fragment-RE term fragment-term from fragment-offset 6-8191
set firewall family inet filter fragment-RE term fragment-term then accept
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure the stateless firewall filter:

1. Define the stateless firewall filter.

```
[edit]
user@host# edit firewall family inet filter fragment-RE
```
2. Configure the first term for the filter.

```
[edit firewall family inet filter fragment-RE ]
user@host# set term not-from-prefix-term from source-address 0.0.0.0/0
user@host# set term not-from-prefix-term from source-address 10.2.1.0/24 except
user@host# set term not-from-prefix-term then discard
```
3. Define the second term for the filter.

```
[edit firewall family inet filter fragment-RE]
user@host# edit term small-offset-term
```
4. Define the match conditions for the term.

```
[edit firewall family inet filter fragment-RE term small-offset-term]
user@host# set from fragment-offset 1-5
```
5. Define the action for the term.

```
[edit firewall family inet filter fragment-RE term small-offset-term]
user@host# set then syslog discard
```
6. Define the third term for the filter.

```
[edit]
user@host# edit firewall family inet filter fragment-RE term not-fragmented-term
```
7. Define the match conditions for the term.

```
[edit firewall family inet filter fragment-RE term not-fragmented-term]
user@host# set from fragment-flags "!more-fragments" fragment-offset 0 protocol
tcp destination-port bgp
```
8. Define the action for the term.

```
[edit firewall family inet filter fragment-RE term not-fragmented-term]
user@host# set then accept
```
9. Define the fourth term for the filter.

```
[edit]
user@host# edit firewall family inet filter fragment-RE term first-fragment-term
```
10. Define the match conditions for the term.

```
[edit firewall family inet filter fragment-RE term first-fragment-term]
user@host# set from first-fragment protocol tcp destination-port bgp
```

11. Define the action for the term.

```
[edit firewall family inet filter fragment-RE term first-fragment-term]
user@host# set then accept
```

12. Define the last term for the filter.

```
[edit]
user@host# edit firewall family inet filter fragment-RE term fragment-term
```

13. Define the match conditions for the term.

```
[edit firewall family inet filter fragment-RE term fragment-term]
user@host# set from fragment-offset 6–8191
```

14. Define the action for the term.

```
[edit firewall family inet filter fragment-RE term fragment-term]
user@host# set then accept
```

Results Confirm your configuration by entering the **show firewall** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show firewall
family inet {
  filter fragment-RE {
    term not-from-prefix-term {
      from {
        source-address {
          0.0.0.0/0;
          10.2.1.0/24 except;
        }
      }
      then discard;
    }
    term small-offset-term {
      from {
        fragment-offset 1-5;
      }
      then {
        syslog;
        discard;
      }
    }
    term not-fragmented-term {
      from {
        fragment-offset 0;
        fragment-flags "!more-fragments";
        protocol tcp;
        destination-port bgp;
      }
      then accept;
    }
    term first-fragment-term {
      from {
        first-fragment;
      }
    }
  }
}
```

```

        protocol tcp;
        destination-port bgp;
    }
    then accept;
}
term fragment-term {
    from {
        fragment-offset 6-8191;
    }
    then accept;
}
}
}

```

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

Verification

To confirm that the configuration is working properly, perform these tasks:

- Displaying Stateless Firewall Filter Configurations on page 165
- Verifying a Firewall Filter that Handles Fragments on page 165

Displaying Stateless Firewall Filter Configurations

Purpose Verify the configuration of the firewall filter. You can analyze the flow of the filter terms by displaying the entire configuration.

Action From configuration mode, enter the **show firewall** command.

Meaning Verify that the output shows the intended configuration of the firewall filter. In addition, verify that the terms are listed in the order in which you want the packets to be tested. You can move terms within a firewall filter by using the **insert** CLI command.

Verifying a Firewall Filter that Handles Fragments

Purpose Verify that the actions of the firewall filter terms are taken.

Action Send packets to the device that match the terms.

Meaning Verify that packets from 10.2.1.0/24 with small fragment offsets are recorded in the device's system logging destinations for the firewall facility.

Related Documentation

- “show route summary” in the *Junos Routing Protocols and Policies Command Reference*.

CHAPTER 15

Standard Firewall Filters for Setting Rate Limits

- Example: Configuring a Rate-Limiting Filter Based on Destination Class on page 167

Example: Configuring a Rate-Limiting Filter Based on Destination Class

This example shows how to configure a rate-limiting stateless firewall filter.

- Requirements on page 167
- Overview on page 167
- Configuration on page 168
- Verification on page 170

Requirements

No special configuration beyond device initialization is required before configuring this example.

Before you begin, configure the destination class **class1**.

Overview

In this example, you use a stateless firewall filter to set rate limits based on a destination class.

To activate a policer from within a stateless firewall filter configuration:

- Create a template for the policer by including the **policer *policer-name*** statement.
- Reference the policer in a filter term that specifies the policer in the **policer *policer-name*** nonterminating action.

You can also activate a policer by including the **policer (input | output) *policer-template-name*** statement at a logical interface.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

- Configure the Stateless Firewall Filter on page 168
- Apply the Stateless Firewall Filter to a Logical Interface on page 168
- Confirm and Commit Your Candidate Configuration on page 169

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall filter rl_dclass1 policer police_class1 if-exceeding bandwidth-limit 25
set firewall filter rl_dclass1 policer police_class1 if-exceeding burst-size-limit 1000
set firewall filter rl_dclass1 policer police_class1 then discard
set firewall filter rl_dclass1 term term1 from destination-class class1
set firewall filter rl_dclass1 term term1 then policer police_class1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.3/30
set interfaces ge-0/0/1 unit 0 family inet filter input ospf_or_131
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter **rl_dclass1** with policer **police_class1** for destination class **class1**:

1. Create the stateless firewall filter **rl_dclass1**.

```
[edit]
user@host# edit firewall filter rl_dclass1
```

2. Configure the policer template **police_class1**.

```
[edit firewall filter rl_dclass1]
user@host# set policer police_class1 if-exceeding bandwidth-limit 25
user@host# set policer police_class1 if-exceeding burst-size-limit 1000
user@host# set policer police_class1 then discard
```

3. Configure a filter term that uses policer **police_class1** to rate-limit traffic for destination class **class1**.

```
[edit firewall filter rl_dclass1]
user@host# set term term1 from destination-class class1
user@host# set term term1 then policer police_class1
```

Apply the Stateless Firewall Filter to a Logical Interface

Step-by-Step Procedure

To apply the filter **rl_dclass1** to a logical interface:

1. Configure the logical interface to which you will apply the stateless firewall filter.

```
[edit]
user@host# edit interfaces ge-0/0/1 unit 0 family inet
```

2. Configure the interface address for the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set address 10.1.2.3/30
```

3. Apply the stateless firewall filter to the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set filter input rl_dclass1
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
filter rl_dclass1 {
  policer police_class1 {
    if-exceeding {
      bandwidth-limit 25;
      burst-size-limit 1000;
    }
    then {
      discard;
    }
  }
}
term term1 {
  from {
    destination-class class1;
  }
  then {
    policer police_class1;
  }
}
}
```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
  unit 0 {
    family inet {
      filter {
        input rl_dclass1;
      }
      address 10.1.2.3/30;
    }
  }
}
```

3. If you are done configuring the device, commit your candidate configuration.

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

Verification

To confirm that the configuration is working properly, enter the **show class-of-service ge-0/0/1** operational mode command.

Related Documentation

- Basic Uses for Standard Firewall Filters on page 24
- Filtering Packets Received on an Interface Set Overview on page 53
- Statement Hierarchy for Defining an Interface Set on page 293
- Statement Hierarchy for Configuring a Filter to Match on an Interface Set on page 293
- Example: Filtering Packets Received on an Interface Set on page 188

CHAPTER 16

Standard Firewall Configuration

- Example: Applying Lists of Multiple Standard Firewall Filters on page 171
- Example: Nesting References to Multiple Standard Firewall Filters on page 176
- Example: Configuring Interface-Specific Firewall Filter Counters on page 180
- Example: Filtering Packets Received on an Interface Group on page 184
- Example: Filtering Packets Received on an Interface Set on page 188
- Example: Configuring Filter-Based Forwarding on the Source Address on page 194

Example: Applying Lists of Multiple Standard Firewall Filters

This example shows how to apply lists of multiple stateless firewall filters.

- Requirements on page 171
- Overview on page 172
- Configuration on page 172
- Verification on page 175

Requirements

Before you begin, be sure that you have:

- Installed your router and supported PIC, DPC, or MPC and performed the initial router configuration.
- Configured basic Ethernet in the topology.
- Configured a logical interface to run the IP version 4 (IPv4) protocol (**family inet**) and configured the logical interface with an interface address. This example uses logical interface **ge-1/3/0.0** configured with the IP address 1.1.1.2/30.



NOTE: For completeness, the configuration section of this example includes setting an IP address for logical interface **ge-1/3/0.0**.

- Verified that traffic is flowing in the topology and that ingress and egress IPv4 traffic is flowing through logical interface **ge-1/3/0.0**.
- Verified that you have access to the remote host that is connected to this router's logical interface **ge-1/3/0.0**.

Overview

In this example, you configure three IPv4 stateless firewall filters and apply each filter directly to the same logical interface by using a list.

Topology

This example applies the following firewall filters as a *list of input filters* at logical interface **ge-1/3/0.0**. Each filter contains a single term that evaluates IPv4 packets and accepts packets based on the value of the **destination port** field in the TCP header:

- Filter **filter_FTP** matches on the FTP port number (**21**).
- Filter **filter_SSH** matches on the SSH port number (**22**).
- Filter **filter_Telnet** matches on the Telnet port number (**23**).

If an inbound packet does not match any of the filters in the input list, the packet is discarded.



NOTE: The Junos OS uses filters in a list in the order in which the filter names appear in the list. In this simple example, the order is irrelevant because all of the filters specify the same action.

Any of the filters can be applied to other interfaces, either alone (using the **input** or **output** statement) or in combination with other filters (using the **input-list** or **output-list** statement). The objective is to configure multiple “minimalist” firewall filters that you can reuse in interface-specific filter lists.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

- Configure Multiple IPv4 Stateless Firewall Filters on page 173
- Apply the Filters to a Logical Interface as an Input List and an Output List on page 174
- Confirm and Commit Your Candidate Configuration on page 174

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter filter_FTP term 0 from protocol tcp
set firewall family inet filter filter_FTP term 0 from destination-port 21
set firewall family inet filter filter_FTP term 0 then count pkts_FTP
set firewall family inet filter filter_FTP term 0 then accept
set firewall family inet filter filter_SSH term 0 from protocol tcp
```

```

set firewall family inet filter filter_SSH term 0 from destination-port 23
set firewall family inet filter filter_SSH term 0 then count pkts_SSH
set firewall family inet filter filter_SSH term 0 then accept
set firewall family inet filter filter_Telnet term 0 from protocol tcp
set firewall family inet filter filter_Telnet term 0 from destination-port 22
set firewall family inet filter filter_Telnet term 0 then count pkts_Telnet
set firewall family inet filter filter_Telnet term 0 then accept
set firewall family inet filter filter_discard term 1 then count pkts_discarded
set firewall family inet filter filter_discard term 1 then discard
set interfaces ge-1/3/0 unit 0 family inet address 1.1.1.2/30
set interfaces ge-1/3/0 unit 0 family inet filter input-list filter_FTP
set interfaces ge-1/3/0 unit 0 family inet filter input-list filter_SSH
set interfaces ge-1/3/0 unit 0 family inet filter input-list filter_Telnet
set interfaces ge-1/3/0 unit 0 family inet filter input-list filter_discard

```

Configure Multiple IPv4 Stateless Firewall Filters

Step-by-Step Procedure

To configure the IPv4 stateless firewall filters:

1. Navigate the CLI to the hierarchy level at which you configure IPv4 firewall filters.

```

[edit]
user@host# edit firewall family inet

```

2. Configure the first firewall filter to count and accept packets for port 21.

```

[edit firewall family inet]
user@host# set filter filter_FTP term 0 from protocol tcp
user@host# set filter filter_FTP term 0 from destination-port 21
user@host# set filter filter_FTP term 0 then count pkts_FTP
user@host# set filter filter_FTP term 0 then accept

```

3. Configure the second firewall filter to count and accept packets for port 23.

```

[edit firewall family inet]
user@host# set filter filter_SSH term 0 from protocol tcp
user@host# set filter filter_SSH term 0 from destination-port 23
user@host# set filter filter_SSH term 0 then count pkt_SSH
user@host# set filter filter_SSH term 0 then accept

```

4. Configure the third firewall filter to count and accept packets from port 22.

```

[edit firewall family inet]
user@host# set filter filter_Telnet term 0 from protocol tcp
user@host# set filter filter_Telnet term 0 from destination-port 22
user@host# set filter filter_Telnet term 0 then count pkts_Telnet
user@host# set filter filter_Telnet term 0 then accept

```

5. Configure the last firewall filter to count the discarded packets.

```

[edit firewall family inet]
user@host# set filter filter_discard term 1 then count pkts_discarded
user@host# set filter filter_discard term 1 then discard

```

Apply the Filters to a Logical Interface as an Input List and an Output List

Step-by-Step Procedure To apply the six IPv4 stateless firewall filters as a list of input filters and a list of output filters:

1. Navigate the CLI to the hierarchy level at which you apply IPv4 firewall filters to logical interface `ge-1/3/0.0`.

[edit]
user@host# **edit interfaces ge-1/3/0 unit 0 family inet**
2. Configure the IPv4 protocol family for the logical interface.

[edit interfaces ge-1/3/0 unit 0 family inet]
user@host# **set address 1.1.1.2/30**
3. Apply the filters as a list of input filters.

[edit interfaces ge-1/3/0 unit 0 family inet]
user@host# **set filter input-list [filter_FTP filter_SSH filter_Telnet filter_discard]**

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filters by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter filter_FTP {
    term 0 {
      from {
        protocol tcp;
        destination-port 21;
      }
      then {
        count pkts_FTP;
        accept;
      }
    }
  }
  filter filter_SSH {
    term 0 {
      from {
        protocol tcp;
        destination-port 23;
      }
      then {
        count pkts_SSH;
        accept;
      }
    }
  }
  filter filter_Telnet {
```



```

term 0 {
  from {
    protocol tcp;
    destination-port 22;
  }
  then {
    count pkts_Telnet;
    accept;
  }
}
filter filter_discard {
  term 1 {
    then {
      count pkts_discarded;
      discard;
    }
  }
}
}

```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

[edit]
user@host# show interfaces
ge-1/3/0 {
  unit 0 {
    family inet {
      filter {
        input-list [ filter_FTP filter_SSH filter_Telnet filter_discard ];
      }
      address 1.1.1.2/30;
    }
  }
}

```

3. If you are done configuring the device, commit your candidate configuration.

```

[edit]
user@host# commit

```

Verification

Confirm that the configuration is working properly.

- Verifying That Inbound Packets Are Accepted Only If Destined for the FTP, SSH or Telnet Port on page 175

Verifying That Inbound Packets Are Accepted Only If Destined for the FTP, SSH or Telnet Port

Purpose Verify that all three filters are active for the logical interface.

Action To verify that input packets are accepted according to the three filters:

1. From the remote host that is connected to this router's logical interface **ge-1/3/0.0**, send a packet with destination port number 21 in the header. The packet should be accepted.
2. From the remote host that is connected to this router's logical interface **ge-1/3/0.0**, send a packet with destination port number 23 in the header. The packet should be accepted.
3. From the remote host that is connected to this router's logical interface **ge-1/3/0.0**, send a packet with destination port number 22 in the header. The packet should be accepted.
4. From the remote host that is connected to this router's logical interface **ge-1/3/0.0**, send a packet with a destination port number *other than* 21, 22, or 23. The packet should be discarded.
5. To display counter information for the list of filters applied to the input at **ge-1/3/0.0-i** enter the **show firewall filter ge-1/3/0.0-i** operational mode command. The command output displays the number of bytes and packets that match filter terms associated with the following counters:
 - **pkts_FTP-ge-1/3/0.0-i**
 - **pkts_SSH-ge-1/3/0.0-i**
 - **pkts_Telnet-ge-1/3/0.0-i**
 - **pkts_discard-ge-1/3/0.0-i**

- Related Documentation**
- Multiple Standard Firewall Filters Applied as a List Overview on page 44
 - Guidelines for Applying Multiple Standard Firewall Filters as a List on page 47

Example: Nesting References to Multiple Standard Firewall Filters

This example shows how to configure nested references to multiple stateless firewall filters.

- Requirements on page 176
- Overview on page 177
- Configuration on page 177
- Verification on page 180

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you configure a stateless firewall filter for a match condition and action combination that can be shared among multiple firewall filters. You then configure two firewall filters that reference the first firewall filter. Later, if the common filtering criteria needs to be changed, you would modify only the one shared firewall filter configuration.

Topology

The **common_filter** firewall filter discards packets that have a UDP source or destination port field number of **69**. Both of the two additional firewall filters, **filter1** and **filter2**, reference the **common_filter**.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see "Using the CLI Editor in Configuration Mode" on page 231.

- Configure the Nested Firewall Filters on page 177
- Apply Both Nested Firewall Filters to Interfaces on page 178
- Confirm and Commit Your Candidate Configuration on page 178

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter common_filter term common_term from protocol udp
set firewall family inet filter common_filter term common_term from port tftp
set firewall family inet filter common_filter term common_term then discard
set firewall family inet filter filter1 term term1 filter common-filter
set firewall family inet filter filter1 term term2 from address 192.168.0.0/16
set firewall family inet filter filter1 term term2 then reject
set firewall family inet filter filter2 term term1 filter common-filter
set firewall family inet filter filter2 term term2 from protocol udp
set firewall family inet filter filter2 term term2 from port bootps
set firewall family inet filter filter2 term term2 then accept
set interfaces ge-0/0/0 unit 0 family inet address 10.1.0.1/24
set interfaces ge-0/0/0 unit 0 family inet filter input filter1
set interfaces ge-0/0/3 unit 0 family inet address 10.1.3.1/24
set interfaces ge-0/0/0 unit 0 family inet filter input filter2
```

Configure the Nested Firewall Filters

Step-by-Step Procedure

To configure two nested firewall filters that share a common filter:

1. Navigate the CLI to the hierarchy level at which you configure IPv4 firewall filters.

```
[edit]
user@host# edit firewall family inet
```

2. Configure the common filter that will be referenced by multiple other filters.

```
[edit firewall family inet]
user@host# set filter common_filter term common_term from protocol udp
user@host# set filter common_filter term common_term from port tftp
```

```
user@host# set filter common_filter term common_term then discard
```

3. Configure a filter that references the common filter.

```
[edit firewall family inet]
user@host# set filter filter1 term term1 filter common-filter
user@host# set filter filter1 term term2 from address 192.168.0.0/16
user@host# set filter filter1 term term2 then reject
```

4. Configure a second filter that references the common filter.

```
[edit firewall family inet]
user@host# set filter filter2 term term1 filter common-filter
user@host# set filter filter2 term term2 from protocol udp
user@host# set filter filter2 term term2 from port bootps
user@host# set filter filter2 term term2 then accept
```

Apply Both Nested Firewall Filters to Interfaces

Step-by-Step Procedure

To apply both nested firewall filters to logical interfaces:

1. Apply the first nested filter to a logical interface input.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.0.1/24
user@host# set interfaces ge-0/0/0 unit 0 family inet filter input filter1
```

2. Apply the second nested filter to a logical interface input.

```
[edit]
user@host# set interfaces ge-0/0/3 unit 0 family inet address 10.1.3.1/24
user@host# set interfaces ge-0/0/0 unit 0 family inet filter input filter2
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter common_filter {
    term common_term {
      from {
        protocol udp;
        port tftp;
      }
      then {
        discard;
      }
    }
  }
}
filter filter1 {
  term term1 {
```

```

        filter common-filter;
    }
    term term2 {
        from {
            address 192.168/16;
        }
        then {
            reject;
        }
    }
}
filter filter2 {
    term term1 {
        filter common-filter;
    }
    term term2 {
        from {
            protocol udp;
            port bootps;
        }
        then {
            accept;
        }
    }
}
}

```

2. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            filter {
                input filter1;
            }
            address 10.1.0.1/24;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            filter {
                input filter2;
            }
            address 10.1.3.1/24;
        }
    }
}
}

```

3. If you are done configuring the device, commit your candidate configuration.

```

[edit]

```

```
user@host# commit
```

Verification

To confirm that the configuration is working properly, enter the **show firewall filter filter1** and **show firewall filter filter2** operational mode commands.

Related Documentation

- Multiple Standard Firewall Filters in a Nested Configuration Overview on page 48
- Guidelines for Nesting References to Multiple Standard Firewall Filters on page 49

Example: Configuring Interface-Specific Firewall Filter Counters

This example shows how to configure and apply an interface-specific standard stateless firewall filter.

- Requirements on page 180
- Overview on page 180
- Configuration on page 181
- Verification on page 183

Requirements

Interface-specific stateless firewall filters are supported on T Series, M120, M320, and MX Series routers only.

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you create an interface-specific stateless firewall filter that counts and accepts packets with source or destination addresses in a specified prefix and the IP protocol type field set to a specific value.

Topology

You configure the interface-specific stateless firewall filter **filter_s_tcp** to count and accept packets with IP source or destination addresses in the **10.0.0.0/12** prefix and the IP protocol type field set to **tcp** (or the numeric value **6**).

The name of the firewall filter counter is **count_s_tcp**.

You apply the firewall filter to multiple logical interfaces:

- **at-1/1/1.0** input
- **so-2/2/2.2** output

Applying the filter to these two interfaces results in two instances of the filter: **filter_s_tcp-at-1/1/1.0-i** and **filter_s_tcp-so-2/2/2.2-o**, respectively.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configure the Interface-Specific Firewall Filter on page 181
- Apply the Interface-Specific Firewall Filter to Multiple Interfaces on page 182
- Confirm Your Candidate Configuration on page 182
- Clear the Counters and Commit Your Candidate Configuration on page 183

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter filter_s_tcp interface-specific
set firewall family inet filter filter_s_tcp term 1 from address 10.0.0.0/12
set firewall family inet filter filter_s_tcp term 1 from protocol tcp
set firewall family inet filter filter_s_tcp term 1 then count count_s_tcp
set firewall family inet filter filter_s_tcp term 1 then accept
set interfaces at-1/1/1 unit 0 family inet filter input filter_s_tcp
set interfaces so-2/2/2 unit 2 family inet filter filter_s_tcp
```

Configure the Interface-Specific Firewall Filter

Step-by-Step Procedure

To configure the interface-specific firewall filter:

1. Create the IPv4 firewall filter **filter_s_tcp**.


```
[edit]
user@host# edit firewall family inet filter filter_s_tcp
```
2. Enable interface-specific instances of the filter.


```
[edit firewall family inet filter filter_s_tcp]
user@host# set interface-specific
```
3. Configure the match conditions for the term.


```
[edit firewall family inet filter filter_s_tcp]
user@host# set term 1 from address 10.0.0.0/12
user@host# set term 1 from protocol tcp
```
4. Configure the actions for the term.


```
[edit firewall family inet filter filter_s_tcp]
user@host# set term 1 then count count_s_tcp
user@host# set term 1 then accept
```

Apply the Interface-Specific Firewall Filter to Multiple Interfaces

Step-by-Step Procedure

To apply the filter **filter_s_tcp** to logical interfaces **at-1/1/1.0** and **so-2/2/2.2**:

1. Apply the interface-specific filter to packets received on logical interface **at-1/1/1.0**.

[edit]
user@host# set interfaces at-1/1/1 unit 0 family inet filter input filter_s_tcp
2. Apply the interface-specific filter to packets transmitted from logical interface **so-2/2/2.2**.

[edit]
user@host# set interfaces so-2/2/2 unit 2 family inet filter filter_s_tcp

Confirm Your Candidate Configuration

Step-by-Step Procedure

To confirm your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter filter_s_tcp {
    interface-specific;
    term 1 {
      from {
        address {
          10.0.0.0/12;
        }
        protocol tcp;
      }
      then {
        count count_s_tcp;
        accept;
      }
    }
  }
}
```

2. Confirm the configuration of the interfaces by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
at-1/1/1 {
  unit 0
    family inet {
      filter {
        input filter_s_tcp;
      }
    }
}
```



```

}
so-2/2/2 {
  unit 2
    family inet {
      filter {
        output filter_s_tcp;
      }
    }
  }
}

```

Clear the Counters and Commit Your Candidate Configuration

Step-by-Step Procedure

To clear the counters and commit your candidate configuration:

1. From operational command mode, use the **clear firewall all** command to clear the statistics for all firewall filters.

To clear only the counters used in this example, include the interface-specific filter instance names:

```

[edit]
user@host> clear firewall filter filter_s_tcp-at-1/1/1.0-i
user@host> clear firewall filter filter_s_tcp-so-2/2/2.2-o

```

2. Commit your candidate configuration.

```

[edit]
user@host# commit

```

Verification

Confirm that the configuration is working properly.

- Verifying That the Filter Is Applied to Each of the Multiple Interfaces on page 183
- Verifying That the Counters Are Collected Separately by Interface on page 184

Verifying That the Filter Is Applied to Each of the Multiple Interfaces

Purpose Verify that the filter is applied to each of the multiple interfaces.

Action Run the **show interfaces** command with the **detail** or **extensive** output level.

1. Verify that the filter is applied to the input for **at-1/1/1.0**:

```

user@host> show interfaces at-1/1/1 detail

```

```

Physical interface: at-1/1/1, Enabled, Physical link is Up
  Interface index: 300, SNMP ifIndex: 194, Generation: 183

```

...

```

Logical interface at-1/1/1.0 (Index 64) (SNMP ifIndex 204) (Generation
5)

```

```

  Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: ATM-SNAP

```

...

```
Protocol inet, MTU: 4470, Generation: 13, Route table: 0
Flags: Sendbroadcast-pkt-to-re
Input Filters: filter_s_tcp-at-1/1/1.0-i,,,,,
```

2. Verify that the filter is applied to the output for **so-2/2/2.2**:

```
user@host> show interfaces so-2/2/2 detail
```

```
Physical interface: so-2/2/2, Enabled, Physical link is Up
Interface index: 129, SNMP ifIndex: 502, Generation: 132
```

```
...
```

```
Logical interface so-2/2/2.2 (Index 70) (SNMP ifIndex 536) (Generation 135)
```

```
Flags: Point-To-Point SNMP-Traps 0x4000 Encapsulation: PPP
```

```
...
```

```
Protocol inet, MTU: 4470, Generation: 146, Route table: 0
Flags: Sendbroadcast-pkt-to-re
Output Filters: filter_s_tcp-so-2/2/2.2-o,,,,,
```

Verifying That the Counters Are Collected Separately by Interface

Purpose Make sure that the **count_s_tcp** counters are collected separately for the two logical interfaces.

Action Run the **show firewall** command.

```
user@host> show firewall filter filter_s_tcp
```

```
Filter: filter_s_tcp
```

```
Counters:
```

Name	Bytes	Packets
count_s_tcp-at-1/1/1.0-i	420	5
count_s_tcp-so-2/2/2.2-o	8888	101

Related Documentation

- [Interface-Specific Firewall Filter Instances Overview on page 51](#)
- [Statement Hierarchy for Configuring Interface-Specific Firewall Filters on page 289](#)
- [Statement Hierarchy for Applying Interface-Specific Firewall Filters on page 290](#)

Example: Filtering Packets Received on an Interface Group

This example shows how to configure a standard stateless firewall filter to match packets tagged for a particular interface group.

- [Requirements on page 185](#)
- [Overview on page 185](#)
- [Configuration on page 185](#)
- [Verification on page 188](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you configure two router interfaces to belong to the interface group. You also configure a stateless firewall filter that matches packets that have been tagged as received on that interface group, contain a source or destination address within a particular prefix, and contain a TCP source or destination port of a specific type and port number. The filter counts, logs, and rejects packets that match this criteria. The filter counts, logs, and rejects all other packets. By applying this firewall filter to only one of the two interfaces in the interface group, you can apply the filtering mechanism to all packets input to the two interfaces.

Topology

You configure the interface group number 1 to consist of the management port **fxp0.0** and the loopback port **lo0.0**.

You configure the firewall filter **filter_if_group** to accept only packets from interface group 1, received from or destined for IP addresses in the 192.168.80.114/32 prefix, and received from or destined for TCP port number 79.

You apply the firewall filter **filter_if_group** to the router's loopback interface.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see "Using the CLI Editor in Configuration Mode" on page 231.

- Configure the Stateless Firewall Filter on page 186
- Assign Interfaces to the Interface Group on page 186
- Apply the Firewall Filter to the Loopback Interface on page 187
- Confirm and Commit Your Candidate Configuration on page 187

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet filter filter_if_group term term1 from interface-group 1
set firewall family inet filter filter_if_group term term1 from address 192.168.80.114/32
set firewall family inet filter filter_if_group term term1 from protocol tcp
set firewall family inet filter filter_if_group term term1 from port finger
set firewall family inet filter filter_if_group term term1 then count if_group_counter1
set firewall family inet filter filter_if_group term term1 then log
set firewall family inet filter filter_if_group term term1 then reject
set firewall family inet filter filter_if_group term term2 then count if_group_counter2
set firewall family inet filter filter_if_group term term2 then log
set firewall family inet filter filter_if_group term term2 then accept
set interfaces fxp0 unit 0 family inet filter group 1
set interfaces fxp0 unit 0 family inet address 192.168.5.38/24
```

```
set interfaces lo0 unit 0 family inet filter group 1
set interfaces lo0 unit 0 family inet address 10.0.0.1/32
set interfaces lo0 unit 0 family inet address 192.168.77.1/32
set interfaces lo0 unit 0 family inet filter input filter_if_group
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter **filter_if_group**:

1. Create the stateless firewall filter **filter_if_group**.

[edit]
user@host# edit firewall family inet filter filter_if_group
2. Configure term **term1** to match packets received on interface group 1, with the source or destination address field in the 192.168.80.114/32 prefix, and with the TCP source or destination port number 79.

[edit firewall family inet filter filter_if_group]
user@host# set term term1 from interface-group 1
user@host# set term term1 from address 192.168.80.114/32
user@host# set term term1 from protocol tcp
user@host# set term term1 from port finger
3. Configure term **term1** to count, log, and reject matching packets.

[edit firewall family inet filter filter_if_group]
user@host# set term term1 then count if_group_counter1
user@host# set term term1 then log
user@host# set term term1 then reject
4. Configure the term **term2** to count, log, and accept all other packets.

[edit firewall family inet filter filter_if_group]
user@host# set term term2 then count if_group_counter2
user@host# set term term2 then log
user@host# set term term2 then accept

Assign Interfaces to the Interface Group

Step-by-Step Procedure

To assign logical interfaces to the interface group number 1 referenced by the firewall filter match term **term1**:

1. Assign the management port to interface group number 1.

[edit]
user@host# set interfaces fxp0 unit 0 family inet filter group 1
user@host# set interfaces fxp0 unit 0 family inet address 192.168.5.38/24
2. Assign a second logical interface to interface group number 1.

[edit]
user@host# set interfaces lo0 unit 0 family inet filter group 1
user@host# set interfaces lo0 unit 0 family inet address 10.0.0.1/32
user@host# set interfaces lo0 unit 0 family inet address 192.168.77.1/32

Apply the Firewall Filter to the Loopback Interface

Step-by-Step Procedure

- To apply the firewall filter to the router's loopback interface:

```
[edit]
user@host# set interfaces lo0 unit 0 family inet filter input filter_if_group
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter filter_if_group {
    term term1 {
      from {
        interface-group 1;
        address {
          192.168.80.114/32;
        }
        protocol tcp;
        port finger;
      }
      then {
        count if_group_counter1;
        log;
        reject;
      }
    }
    term term2 {
      then {
        count if_group_counter2;
        log;
        accept;
      }
    }
  }
}
```

2. Confirm the configuration of the interfaces by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
fxp0 {
  unit 0 {
    family inet {
      filter {
        group 1;
      }
    }
  }
}
```

```
        address 192.168.5.38/24;
    }
}
lo0 {
    unit 0 {
        family inet {
            filter {
                input filter_if_group;
                group 1;
            }
            address 10.0.0.1/32;
            address 192.168.77.1/32;
        }
    }
}
```

3. If you are done configuring the device, commit your candidate configuration.

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

Verification

To display the firewall filter counters, enter the **show firewall filter filter_if_group** operational mode command:

```
user@host> show firewall filter filter_if_group
```

```
Filter: filter_if_group
```

```
Counters:
```

Name	Bytes	Packets
if_group_counter1	0	0
if_group_counter2	6452105	82667

To display the local log of packet headers for packets evaluated by the firewall filter, enter the **show firewall log** operational mode command.

Related Documentation

- Filtering Packets Received on a Set of Interface Groups Overview on page 52
- Statement Hierarchy for Assigning Interfaces to Interface Groups on page 291
- Statement Hierarchy for Configuring a Filter to Match on a Set of Interface Groups on page 291

Example: Filtering Packets Received on an Interface Set

This example shows how to configure a standard stateless firewall filter to match packets tagged for a particular interface set.

- Requirements on page 189
- Overview on page 189
- Configuration on page 189
- Verification on page 194

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you apply a stateless firewall filter to the input of the router loopback interface. The firewall filter includes a term that matches packets tagged for a particular interface set.

Topology

You create the firewall filter **L2_filter** to apply rate limits to the protocol-independent traffic received on the following interfaces:

- **fe-0/0/0.0**
- **fe-1/0/0.0**
- **fe-1/1/0.0**

First, for protocol-independent traffic received on **fe-0/0/0.0**, the firewall filter term **t1** applies policer **p1**.

For protocol-independent traffic received on any other Fast Ethernet interfaces, firewall filter term **t2** applies policer **p2**. To define an interface set that consists of all Fast Ethernet interfaces, you include the **interface-set *interface-set-name interface-name*** statement at the **[edit firewall]** hierarchy level. To define a packet-matching criteria based on the interface on which a packet arrives to a specified interface set, you configure a term that uses the **interface-set** firewall filter match condition.

Finally, for any other protocol-independent traffic, firewall filter term **t3** applies policer **p3**.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configuring the Interfaces for Which the Stateless Firewall Filter Terms Take Rate-Limiting Actions on page 190
- Configuring the Stateless Firewall Filter That Rate-Limits Protocol-Independent Traffic Based on the Interfaces on Which Packets Arrive on page 191
- Applying the Stateless Firewall Filter to the Routing Engine Input Interface on page 193

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set interfaces fe-0/0/0 unit 0 family inet address 10.1.1/30
```

```
set interfaces fe-1/0/0 unit 0 family inet address 10.2.2.1/30
set interfaces fe-1/1/0 unit 0 family inet address 10.4.4.1/30
set firewall policer p1 if-exceeding bandwidth-limit 5m
set firewall policer p1 if-exceeding burst-size-limit 10m
set firewall policer p1 then discard
set firewall policer p2 if-exceeding bandwidth-limit 40m
set firewall policer p2 if-exceeding burst-size-limit 100m
set firewall policer p2 then discard
set firewall policer p3 if-exceeding bandwidth-limit 600m
set firewall policer p3 if-exceeding burst-size-limit 1g
set firewall policer p3 then discard
set firewall interface-set ifset fe-*
set firewall family any filter L2_filter term t1 from interface fe-0/0/0.0
set firewall family any filter L2_filter term t1 then count c1
set firewall family any filter L2_filter term t1 then policer p1
set firewall family any filter L2_filter term t2 from interface-set ifset
set firewall family any filter L2_filter term t2 then count c2
set firewall family any filter L2_filter term t2 then policer p2
set firewall family any filter L2_filter term t3 then count c3
set firewall family any filter L2_filter term t3 then policer p3
set interfaces lo0 unit 0 family inet address 1.1.1.157/30
set interfaces lo0 unit 0 filter input L2_filter
```

Configuring the Interfaces for Which the Stateless Firewall Filter Terms Take Rate-Limiting Actions

Step-by-Step Procedure

To configure the interfaces for which the stateless firewall filter terms take rate-limiting actions:

1. Configure the logical interface whose input traffic will be matched by the first term of the firewall filter.

```
[edit]
user@host# set interfaces fe-0/0/0 unit 0 family inet address 10.1.1.1/30
```

2. Configure the logical interfaces whose input traffic will be matched by the second term of the firewall filter.

```
[edit ]
user@host# set interfaces fe-1/0/0 unit 0 family inet address 10.2.2.1/30
user@host# set interfaces fe-1/1/0 unit 0 family inet address 10.4.4.1/30
```

3. If you are done configuring the device, commit the configuration.

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

Results Confirm the configuration of the router transit interfaces by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show interfaces
fe-0/0/0 {
  unit 0 {
    family inet {
      address 10.1.1.1/30;
```



```

    }
  }
}
fe-1/0/0 {
  unit 0 {
    family inet {
      address 10.2.2.1/30;
    }
  }
}
fe-1/1/0 {
  unit 0 {
    family inet {
      address 10.4.4.1/30;
    }
  }
}
}

```

Configuring the Stateless Firewall Filter That Rate-Limits Protocol-Independent Traffic Based on the Interfaces on Which Packets Arrive

Step-by-Step Procedure

To configure the standard stateless firewall **L2_filter** that uses policers (**p1**, **p2**, and **p3**) to rate-limit protocol-independent traffic based on the interfaces on which the packets arrive:

1. Configure the firewall statements.

```

[edit]
user@host# edit firewall

```

2. Configure the policer **p1** to discard traffic that exceeds a traffic rate of **5m** bps or a burst size of **10m** bytes.

```

[edit firewall]
user@host# set policer p1 if-exceeding bandwidth-limit 5m
user@host# set policer p1 if-exceeding burst-size-limit 10m
user@host# set policer p1 then discard

```

3. Configure the policer **p2** to discard traffic that exceeds a traffic rate of **40m** bps or a burst size of **100m** bytes.

```

[edit firewall]
user@host# set policer p2 if-exceeding bandwidth-limit 40m
user@host# set policer p2 if-exceeding burst-size-limit 100m
user@host# set policer p2 then discard

```

4. Configure the policer **p3** to discard traffic that exceeds a traffic rate of **600m** bps or a burst size of **1g** bytes.

```

[edit firewall]
user@host# set policer p3 if-exceeding bandwidth-limit 600m
user@host# set policer p3 if-exceeding burst-size-limit 1g
user@host# set policer p3 then discard

```

5. Define the interface set **ifset** to be the group of all Fast Ethernet interfaces on the router.

```

[edit firewall]

```

```
user@host# set interface-set ifset fe-*
```

6. Create the stateless firewall filter **L2_filter**.

```
[edit firewall]
user@host# edit family any filter L2_filter
```

7. Configure filter term **t1** to match IPv4, IPv6, or MPLS packets received on interface **fe-0/0/0.0** and use policer **p1** to rate-limit that traffic.

```
[edit firewall family any filter L2_filter]
user@host# set term t1 from interface fe-0/0/0.0
user@host# set term t1 then count c1
user@host# set term t1 then policer p1
```

8. Configure filter term **t2** to match packets received on interface-set **ifset** and use policer **p2** to rate-limit that traffic.

```
[edit firewall family any filter L2_filter]
user@host# set term t2 from interface-set ifset
user@host# set term t2 then count c2
user@host# set term t2 then policer p2
```

9. Configure filter term **t3** to use policer **p3** to rate-limit all other traffic.

```
[edit firewall family any filter L2_filter]
user@host# set term t3 then count c3
user@host# set term t3 then policer p3
```

10. If you are done configuring the device, commit the configuration.

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

Results Confirm the configuration of the stateless firewall filter and the policers referenced as firewall filter actions by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show firewall
family any {
  filter L2_filter {
    term t1 {
      from {
        interface fe-0/0/0.0;
      }
      then {
        policer p1;
        count c1;
      }
    }
    term t2 {
      from {
        interface-set ifset;
      }
      then {
        policer p2;
      }
    }
  }
}
```

```

        count c2;
    }
}
term t3 {
    then {
        policer p3;
        count c3;
    }
}
}
}
policer p1 {
    if-exceeding {
        bandwidth-limit 5m;
        burst-size-limit 10m;
    }
    then discard;
}
policer p2 {
    if-exceeding {
        bandwidth-limit 40m;
        burst-size-limit 100m;
    }
    then discard;
}
policer p3 {
    if-exceeding {
        bandwidth-limit 600m;
        burst-size-limit 1g;
    }
    then discard;
}
interface-set ifset {
    fe-*;
}

```

Applying the Stateless Firewall Filter to the Routing Engine Input Interface

Step-by-Step Procedure

To apply the stateless firewall filter to the Routing Engine input interface:

1. Apply the stateless firewall filter to the Routing Engine interface in the input direction.

[edit]

```

user@host# set interfaces lo0 unit 0 family inet address 1.1.1.157/30
user@host# set interfaces lo0 unit 0 filter input L2_filter

```

2. If you are done configuring the device, commit the configuration.

[edit]

```

user@host# commit

```

Results Confirm the application of the firewall filter to the Routing Engine input interface by entering the **show interfaces** command again. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
user@host# show interfaces
fe-0/0/0 {
  ...
}
fe-1/0/0 {
  ...
}
fe-1/1/0 {
  ...
}
lo0 {
  unit 0 {
    filter {
      input L2_filter;
    }
    family inet {
      address 1.1.1.157/30;
    }
  }
}
```

Verification

To confirm that the configuration is working properly, use the **show firewall filter L2_filter** operational mode command to monitor traffic statistics about the firewall filter and three counters.

- Related Documentation**
- Basic Uses for Standard Firewall Filters on page 24
 - Filtering Packets Received on an Interface Set Overview on page 53
 - Statement Hierarchy for Defining an Interface Set on page 293
 - Statement Hierarchy for Configuring a Filter to Match on an Interface Set on page 293

Example: Configuring Filter-Based Forwarding on the Source Address

This example shows how to configure filter-based forwarding (FBF).

- Requirements on page 194
- Overview on page 195
- Configuration on page 195
- Verification on page 198

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, the router receives IPv4 traffic from one application server that is destined for a different application server. The router matches this flow of data packets using a firewall filter based on the packet IP source address. Any matching packets are routed to a routing instance that first sends all traffic to a security device, then forwards the traffic to the designated destination address.

Topology

The source application server is at host IP address 10.1.0.1/24, and the destination application server is at host IP address 10.1.3.1/24.

The router interface to the source application server is **ge-0/0/0.0**, and the router interface to the destination application server is **ge-0/0/3.0**.

The routing instance **ri_fbf** routing table **ri_fbf.inet0** is configured with the interface **ge-0/0/1.0** to the security device and the interface **ge-0/0/3.0** to the destination application server.

The firewall filter **filter_fbf** matches packets on the source IP address 1.1.1.1/32 and directs matching packets to the routing instance **ri_fbf**,

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configure the Routing Instance on page 196
- Configure the Stateless Firewall Filter on page 196
- Configure the Interfaces to the Application Servers and Apply the Stateless Firewall Filter on page 196
- Confirm and Commit Your Candidate Configuration on page 197

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.0.1/24
set interfaces ge-0/0/3 unit 0 family inet address 10.1.3.1/24
set routing-instances ri_fbf instance-type forwarding
set routing-instances ri_fbf interface ge-0/0/1.0
set routing-instances ri_fbf interface ge-0/0/3.0
set routing-instances ri_fbf routing-options static route 12.34.56.0/24 next-hop 10.1.3.254
set firewall family inet filter filter_fbf term t1 from protocol tcp
set firewall family inet filter filter_fbf term t1 from source-address 1.1.1.1/32
set firewall family inet filter filter_fbf term t1 then routing-instance ri_fbf
set interfaces ge-0/0/0 unit 0 family inet filter input filter_fbf
```

Configure the Routing Instance

Step-by-Step Procedure

To configure the routing instance:

1. Create the routing instance. The **forwarding** routing instance type supports filter-based forwarding, where interfaces are not associated with instances. For this instance type, there is no one-to-one mapping between an interface and a routing instance. All interfaces belong to the default instance **inet.0**.

```
[edit]
user@host# set routing-instances ri_fbf instance-type forwarding
```

2. Associate the interfaces with the routing instance.

```
[edit]
user@host# set routing-instances ri_fbf interface ge-0/0/1.0
user@host# set routing-instances ri_fbf interface ge-0/0/3.0
```

3. Configure the routing information for the routing instance.

```
[edit]
user@host# set routing-instances ri_fbf routing-options static route 12.34.56.0/24
next-hop 10.1.3.254
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter:

1. Create the firewall filter.

```
[edit]
user@host# set firewall family inet filter filter_fbf term t1 from protocol tcp
```

2. Set the firewall filter to match the correct source address.

```
[edit]
user@host# set firewall family inet filter filter_fbf term t1 from source-address
1.1.1.1/32
```

3. Set the firewall filter to forward packets to the routing instance.

```
[edit]
user@host# set firewall family inet filter filter_fbf term t1 then routing-instance ri_fbf
```

Configure the Interfaces to the Application Servers and Apply the Stateless Firewall Filter

Step-by-Step Procedure

To configure the interfaces to the application servers and apply the stateless firewall filter to the interface to the source application server:

1. Configure the interface to the source application server.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.0.1/24
```

2. Configure the interface to the destination application server.

```
[edit]
```

```
user@host# set interfaces ge-0/0/3 unit 0 family inet address 10.1.3.1/24
```

3. Apply the stateless firewall filter to the input from the source application server.

```
[edit]
```

```
user@host# set interfaces ge-0/0/0 unit 0 family inet filter input filter_fbf
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the routing instance by entering the **show routing-instances** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
```

```
user@host# show routing-instances
```

```
ri_fbf {
  instance-type forwarding;
  interface ge-0/0/1.0;
  interface ge-0/0/3.0;
  routing-options {
    static {
      route 12.34.56.0/24 next-hop 10.1.3.254;
    }
  }
}
```

2. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
```

```
user@host# show firewall
```

```
family inet {
  filter filter_fbf {
    term t1 {
      from {
        source-address {
          1.1.1.1/32;
        }
        protocol tcp;
      }
      then {
        routing-instance ri_fbf;
      }
    }
  }
}
```

3. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
```

```
user@host# show interfaces
```

```

ge-0/0/0 {
  unit 0 {
    family inet {
      filter {
        input filter_fbf;
      }
      address 10.1.0.1/24;
    }
  }
}
ge-0/0/3 {
  unit 0 {
    family inet {
      address 10.1.3.1/24;
    }
  }
}

```

Verification

Confirm that the configuration is working properly.

Verifying That Filter-Based Forwarding Is Configured

Purpose Verify that the firewall filter is enabled on the interface to the source application server and that the router forwarding table and the routing instance forwarding table contain the correct information.

Action To perform the verification:

1. Use the **show interfaces filters** command to display firewall filter information for the interface to the source application server.

```
user@host> show interfaces filters ge-0/0/0.0
```

Interface	Admin	Link	Proto	Input Filter	Output Filter
ge-0/0/0.0	up	down	inet	filter_fbf	

2. Use the **show route forwarding-table** command to verify the contents of the router forwarding table and the routing instance forwarding table.

```
user@host> show route forwarding-table
```

Routing table: default.inet

Internet:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	user	1	0:12:f2:21:cf:0	ucst	331	4	me0.0
default	perm	0		rjct	36	3	
0.0.0.0/32	perm	0		dscd	34	1	
10.1.0.0/24	ifdn	0		rsrv	613	1	
ge-0/0/0.0							
10.1.0.0/32	iddn	0	10.1.0.0	recv	611	1	
ge-0/0/0.0							
10.1.0.1/32	user	0		rjct	36	3	
10.1.0.1/32	intf	0	10.1.0.1	loc1	612	2	
10.1.0.1/32	iddn	0	10.1.0.1	loc1	612	2	
10.1.0.255/32	iddn	0	10.1.0.255	bcst	610	1	


```

ge-0/0/0.0
10.1.1.0/26      ifdn    0          rslv    583    1 vlan.0
10.1.1.0/32      iddn    0 10.1.1.0  recv    581    1 vlan.0
10.1.1.1/32      user    0          rjct     36     3
10.1.1.1/32      intf    0 10.1.1.1  locl    582     2
10.1.1.1/32      iddn    0 10.1.1.1  locl    582     2
10.1.1.63/32     iddn    0 10.1.1.63 bcst    580    1 vlan.0
255.255.255.255/32 perm    0          bcst     32     1

```

Routing table: ri_fbf.inet

Internet:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	559	2	
0.0.0.0/32	perm	0		dscd	545	1	
10.1.3.0/24	ifdn	0		rslv	617	1	
ge-0/0/3.0							
10.1.3.0/32	iddn	0 10.1.3.0		recv	615	1	
ge-0/0/3.0							
10.1.3.1/32	user	0		rjct	559	2	
10.1.3.1/32	intf	0 10.1.3.1		locl	616	2	
10.1.3.1/32	iddn	0 10.1.3.1		locl	616	2	
10.1.3.255/32	iddn	0 10.1.3.255		bcst	614	1	
ge-0/0/3.0							
224.0.0.0/4	perm	0		mdsc	546	1	
224.0.0.1/32	perm	0 224.0.0.1		mcst	529	1	
255.255.255.255/32	perm	0		bcst	543	1	

Routing table: default.iso

ISO:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	60	1	

Routing table: ri_fbf.iso

ISO:

Destination	Type	RtRef	Next hop	Type	Index	NhRef	Netif
default	perm	0		rjct	600	1	

Meaning The output indicates that the filter was created on the interface and that the routing instance is forwarding matching traffic to the correct IP address.

- Related Documentation**
- Filter-Based Forwarding Overview on page 53
 - Statement Hierarchy for Configuring FBF for IPv4 or IPv6 Traffic on page 294
 - Statement Hierarchy for Configuring FBF for MPLS-Tagged IPv4 Traffic on page 295
 - Statement Hierarchy for Configuring Routing Instances for FBF on page 297
 - Statement Hierarchy for Applying FBF Filters to Interfaces on page 298

Standard Firewall Configuration Options

- Example: Configuring Statistics Collection for a Standard Firewall Filter on page 201
- Example: Configuring Logging for a Stateless Firewall Filter Term on page 206

Example: Configuring Statistics Collection for a Standard Firewall Filter

This example shows how to configure and apply a stateless firewall filter that collects data according to parameters specified in an associated accounting profile.

- Requirements on page 201
- Overview on page 201
- Configuration on page 202
- Verification on page 205

Requirements

Firewall filter accounting profiles are supported for all traffic types except **family any**.

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you create a firewall filter accounting profile and apply it to a stateless firewall filter. The accounting profile specifies how frequently to collect packet and byte count statistics and the name of the file to which the statistics are written. The profile also specifies that statistics are to be collected for three firewall filter counters.

Topology

The firewall filter accounting profile **filter_acctg_profile** specifies that statistics are collected every 60 minutes, and the statistics are written to the file **/var/log/ff_accounting_file**. Statistics are collected for counters named **counter1**, **counter2**, and **counter3**.

The IPv4 stateless firewall filter named **my_firewall_filter** increments a counter for each of three filter terms. The filter is applied to logical interface **ge-0/0/1.0**.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configure an Accounting Profile on page 202
- Configure a Firewall Filter That References the Accounting Profile on page 203
- Apply the Firewall Filter to an Interface on page 203
- Confirm Your Candidate Configuration on page 204
- Clear the Counters and Commit Your Candidate Configuration on page 205

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the [edit] hierarchy level.

```
set accounting-options filter-profile filter_acctg_profile file ff_accounting_file
set accounting-options filter-profile filter_acctg_profile interval 60
set accounting-options filter-profile filter_acctg_profile counters counter1
set accounting-options filter-profile filter_acctg_profile counters counter2
set accounting-options filter-profile filter_acctg_profile counters counter3
set firewall family inet filter my_firewall_filter accounting-profile filter_acctg_profile
set firewall family inet filter my_firewall_filter term term1 from protocol ospf
set firewall family inet filter my_firewall_filter term term1 then count counter1
set firewall family inet filter my_firewall_filter term term1 then discard
set firewall family inet filter my_firewall_filter term term2 from source-address
  10.108.0.0/16
set firewall family inet filter my_firewall_filter term term2 then count counter2
set firewall family inet filter my_firewall_filter term term2 then discard
set firewall family inet filter my_firewall_filter term accept-all then count counter3
set firewall family inet filter my_firewall_filter term accept-all then accept
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.3/30
set interfaces ge-0/0/1 unit 0 family inet filter input my_firewall_filter
```

Configure an Accounting Profile

Step-by-Step Procedure

To configure an accounting profile:

1. Create the accounting profile **filter_acctg_profile**.

```
[edit]
user@host# edit accounting-options filter-profile filter_acctg_profile
```

2. Configure the accounting profile to filter and collect packet and byte count statistics every 60 minutes and write them to the **/var/log/ff_accounting_file** file.

```
[edit accounting-options filter-profile filter_acctg_profile]
user@host# set file ff_accounting_file
user@host# set interval 60
```

3. Configure the accounting profile to collect filter profile statistics (packet and byte counts) for three counters.

```
[edit accounting-options filter-profile filter_acctg_profile]
user@host# set counters counter1
user@host# set counters counter2
user@host# set counters counter3
```

Configure a Firewall Filter That References the Accounting Profile

Step-by-Step Procedure

To configure a firewall filter that references the accounting profile:

1. Create the stateless firewall filter **my_firewall_filter**.

```
[edit]
user@host# edit firewall family inet filter my_firewall_filter
```

2. Apply the filter-accounting profile **filter_acctg_profile** to the firewall filter.

```
[edit firewall family inet filter my_firewall_filter]
user@host# set accounting-profile filter_acctg_profile
```

3. Configure the first filter term and counter.

```
[edit firewall family inet filter my_firewall_filter]
user@host# set term term1 from protocol ospf
user@host# set term term1 then count counter1
user@host# set term term1 then discard
```

4. Configure the second filter term and counter.

```
[edit firewall family inet filter my_firewall_filter]
user@host# set term term2 from source-address 10.108.0.0/16
user@host# set term term2 then count counter2
user@host# set term term2 then discard
```

5. Configure the third filter term and counter.

```
[edit firewall family inet filter my_firewall_filter]
user@host# set term accept-all then count counter3
user@host# set term accept-all then accept
```

Apply the Firewall Filter to an Interface

Step-by-Step Procedure

To apply the stateless firewall filter to a logical interface:

1. Configure the logical interface to which you will apply the stateless firewall filter.

```
[edit]
user@host# edit interfaces ge-0/0/1 unit 0 family inet
```

2. Configure the interface address for the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set address 10.1.2.3/30
```

3. Apply the stateless firewall filter to the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
```

```
user@host# set filter input my_firewall_filter
```

Confirm Your Candidate Configuration

Step-by-Step Procedure

To confirm your candidate configuration:

1. Confirm the configuration of the accounting profile by entering the **show accounting-options** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show accounting-options
filter-profile filter_acctg_profile {
  file ff_accounting_file;
  interval 60;
  counters {
    counter1;
    counter2;
    counter3;
  }
}
```

2. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter my_firewall_filter {
    accounting-profile filter_acctg_profile;
    term term1 {
      from {
        protocol ospf;
      }
      then {
        count counter1;
        discard;
      }
    }
    term term2 {
      from {
        source-address {
          10.108.0.0/16;
        }
      }
      then {
        count counter2;
        discard;
      }
    }
    term accept-all {
      then {
        count counter3;
        accept;
      }
    }
  }
}
```

```

    }
  }
}

```

3. Confirm the configuration of the interfaces by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

[edit]
user@host# show interfaces
ge-0/0/1 {
  unit 0 {
    family inet {
      filter {
        input my_firewall_filter;
      }
      address 10.1.2.3/30;
    }
  }
}

```

Clear the Counters and Commit Your Candidate Configuration

Step-by-Step Procedure

To clear the counters and commit your candidate configuration:

1. From operational command mode, use the **clear firewall all** command to clear the statistics for all firewall filters.

To clear only the counters incremented in this example, include the name of the firewall filter.

```

[edit]
user@host> clear firewall filter my_firewall_filter

```

2. Commit your candidate configuration.

```

[edit]
user@host# commit

```

Verification

To verify that the filter is applied to the logical interface, run the **show interfaces** command with the **detail** or **extensive** output level.

To verify that the three counters are collected separately, run the **show firewall filter my_firewall_filter** command.

```

user@host> show firewall filter my_firewall_filter

```

```

Filter: my_firewall_filter
Counters:
Name                               Bytes      Packets
counter1                           0           0
counter2                           0           0

```

counter3

0

0

**Related
Documentation**

- Accounting for Standard Firewall Filters Overview on page 55
- Statement Hierarchy for Configuring Firewall Filter Accounting Profiles on page 299
- Statement Hierarchy for Applying Firewall Filter Accounting Profiles on page 300

Example: Configuring Logging for a Stateless Firewall Filter Term

This example shows how to configure a standard stateless firewall filter to log packet headers.

- Requirements on page 206
- Overview on page 206
- Configuration on page 206
- Verification on page 209

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, you use a stateless firewall filter that logs and counts ICMP packets that have 192.168.207.222 as either their source or destination.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see "Using the CLI Editor in Configuration Mode" on page 231.

To configure this example, perform the following tasks:

- Configure the Syslog Messages File for the Firewall Facility on page 207
- Configure the Stateless Firewall Filter on page 207
- Apply the Stateless Firewall Filter to a Logical Interface on page 207
- Confirm and Commit Your Candidate Configuration on page 208

**CLI Quick
Configuration**

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set system syslog file messages_firewall_any firewall any
set system syslog file messages_firewall_any archive no-world-readable
set firewall family inet filter icmp_syslog term icmp_match from address
  192.168.207.222/32
set firewall family inet filter icmp_syslog term icmp_match from protocol icmp
set firewall family inet filter icmp_syslog term icmp_match then count packets
```



```

set firewall family inet filter icmp_syslog term icmp_match then log
set firewall family inet filter icmp_syslog term icmp_match then accept
set firewall family inet filter icmp_syslog term default_term then accept
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.3/30
set interfaces ge-0/0/1 unit 0 family inet filter input icmp_syslog

```

Configure the Syslog Messages File for the Firewall Facility

Step-by-Step Procedure

To configure a syslog messages file for the **firewall** facility:

1. Configure a messages file for all syslog messages generated for the **firewall** facility.

```
user@host# set system syslog file messages_firewall_any firewall any
```

2. Restrict permission to the archived **firewall** facility syslog files to the root user and users who have the Junos OS maintenance permission.

```
user@host# set system syslog file messages_firewall_any archive no-world-readable
```

Configure the Stateless Firewall Filter

Step-by-Step Procedure

To configure the stateless firewall filter **icmp_syslog** that logs and counts ICMP packets that have **192.168.207.222** as either their source or destination:

1. Create the stateless firewall filter **icmp_syslog**.

```
[edit]
user@host# edit firewall family inet filter icmp_syslog
```

2. Configure matching on the ICMP protocol and an address.

```
[edit firewall family inet filter icmp_syslog]
user@host# set term icmp_match from address 192.168.207.222/32
user@host# set term icmp_match from protocol icmp
```

3. Count, log, and accept matching packets.

```
[edit firewall family inet filter icmp_syslog]
user@host# set term icmp_match then count packets
user@host# set term icmp_match then log
user@host# set term icmp_match then accept
```

4. Accept all other packets.

```
[edit firewall family inet filter icmp_syslog]
user@host# set term default_term then accept
```

Apply the Stateless Firewall Filter to a Logical Interface

Step-by-Step Procedure

To apply the stateless firewall filter to a logical interface:

1. Configure the logical interface to which you will apply the stateless firewall filter.

```
[edit]
user@host# edit interfaces ge-0/0/1 unit 0 family inet
```

2. Configure the interface address for the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
```

```
user@host# set address 10.1.2.3/30
```

3. Apply the stateless firewall filter to the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set filter input icmp_syslog
```

Confirm and Commit Your Candidate Configuration

Step-by-Step Procedure

To confirm and then commit your candidate configuration:

1. Confirm the configuration of the syslog message file for the **firewall** facility by entering the **show system** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show system
syslog {
  file messages_firewall_any {
    firewall any;
    archive no-world-readable;
  }
}
```

2. Confirm the configuration of the stateless firewall filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  filter icmp_syslog {
    term icmp_match {
      from {
        address {
          192.168.207.222/32;
        }
        protocol icmp;
      }
      then {
        count packets;
        log;
        accept;
      }
    }
    term default_term {
      then accept;
    }
  }
}
```

3. Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
```

```

user@host# show interfaces
ge-0/0/1 {
  unit 0 {
    family inet {
      filter {
        input icmp_syslog;
      }
      address 10.1.2.3/30;
    }
  }
}

```

4. If you are done configuring the device, commit your candidate configuration.

```

[edit]
user@host# commit

```

Verification

To confirm that the configuration is working properly, enter the **show log filter** command:

```

user@host> show log messages_firewall_any
Mar 20 08:03:11 hostname feb FW: so-0/1/0.0   A icmp 192.168.207.222
192.168.207.223      0      0 (1 packets)

```

This output file contains the following fields:

- **Date and Time**—Date and time at which the packet was received (not shown in the default).
- **Filter action**:
 - **A**—Accept (or next term)
 - **D**—Discard
 - **R**—Reject
- **Protocol**—Packet's protocol name or number.
- **Source address**—Source IP address in the packet.
- **Destination address**—Destination IP address in the packet.



NOTE: If the protocol is ICMP, the ICMP type and code are displayed. For all other protocols, the source and destination ports are displayed.

The last two fields (both zero) are the source and destination TCP/UDP ports, respectively, and are shown for TCP or UDP packets only. This log message indicates that only one packet for this match has been detected in about a 1-second interval. If packets arrive faster, the system log function compresses the information so that less output is generated, and displays an output similar to the following:

```

user@host> show log filter
Mar 20 08:08:45 hostname feb FW: so-0/1/0.0   A icmp 192.168.207.222

```

192.168.207.223 0 0 (515 packets)

**Related
Documentation**

- System Logging Overview on page 55
- Logging of Packet Headers Evaluated by a Firewall Filter Term on page 58
- System log messages with the **DFWD_** prefix, described in the [Junos OS System Log Messages Reference](#)
- System log messages with the **PFE_FW_*** prefix, described in the [Junos OS System Log Messages Reference](#)

CHAPTER 18

Service Filter Configuration

- Example: Configuring and Applying Service Filters on page 211

Example: Configuring and Applying Service Filters

This example shows how to configure and apply service filters.

- Requirements on page 211
- Overview on page 211
- Configuration on page 212
- Verification on page 215

Requirements

This example use the logical interface **xe-0/1/0.0** on any of the following hardware components:

- Adaptive Services (AS) PIC on an M Series or T Series router
- Multiservices (MS) PIC on an M Series or T Series router
- Multiservices (MS) DPC on an MX Series router

Before you begin, make sure that you have:

- Installed your supported router and PICs or DPCs and performed the initial router configuration.
- Configured basic Ethernet in the topology, and verified that traffic is flowing in the topology and that IPv4 traffic is flowing through logical interface **xe-0/1/0.0**.
- Configured the service set **vrf_svcs** with service input and output rules and default settings for services at a service interface.

For guidelines for configuring service sets, see “Configuring Service Sets to be Applied to Services Interfaces” in the *Junos OS Services Interfaces Configuration Guide*.

Overview

In this example, you create three types of service filters for IPv4 traffic: one input service filter, one postservice input filter, and one output service filter.

Topology

You apply the input service filter and postservice input filter to input traffic at logical interface **xe-0/1/0.0**, and you apply the output service filter to the output traffic at the same logical interface.

- Filtering IPv4 traffic before it is accepted for input service processing—At logical interface **xe-0/1/0.0**, you use the service filter **in_filter_presvc** to filter IPv4 input traffic before the traffic can be accepted for processing by services associated with service set **vrf_svcs**. The **in_filter_presvc** service filter counts packets sent from ICMP port 179, directs these packets to the input services associated with the service set **vrf_svcs**, and discards all other packets.
- Filtering IPv4 traffic after it has completed input service processing—At logical interface **xe-0/1/0.0**, you use the service filter **in_filter_postsvc** to filter traffic that is returning to the services interface after the input service set **in_filter_presvc** is executed. The **in_filter_postsvc** service filter counts packets sent from ICMP port 179 and then discards them.
- Filtering IPv4 traffic before it is accepted for output service processing—At logical interface **xe-0/1/0.0**, you use the service-filter **out_filter_presvc** to filter IPv4 output traffic before the traffic can be accepted for processing by the services associated with service set **vrf_svcs**. The **out_filter_presvc** service filter counts packets destined for TCP port 179 and then directs the packets to the output services associated with the service set **vrf_svcs**.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configuring the Three Service Filters on page 213
- Applying the Three Service Filters on page 214

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet service-filter in_filter_presvc term t1 from protocol tcp
set firewall family inet service-filter in_filter_presvc term t1 from source-port bgp
set firewall family inet service-filter in_filter_presvc term t1 then count svc_in_pkts
set firewall family inet service-filter in_filter_postsvc term t2 from protocol tcp
set firewall family inet service-filter in_filter_postsvc term t2 from source-port bgp
set firewall family inet service-filter in_filter_postsvc term t2 then count svc_in_pkts_rtn
set firewall family inet service-filter in_filter_postsvc term t2 then skip
set firewall family inet service-filter out_filter_presvc term t3 from protocol icmp
set firewall family inet service-filter out_filter_presvc term t3 from destination-port bgp
set firewall family inet service-filter out_filter_presvc term t3 then count svc_out_pkts
set firewall family inet service-filter out_filter_presvc term t3 then service
```

```

set interfaces xe-0/1/0 unit 0 family inet service input service-set vrf_svcs service-filter
in_filter_presvc
set interfaces xe-0/1/0 unit 0 family inet service input post-service-filter in_filter_postsvc
set interfaces xe-0/1/0 unit 0 family inet service output service-set vrf_svcs service-filter
out_filter_presvc

```

Configuring the Three Service Filters

Step-by-Step Procedure

To configure the three service filters:

1. Configure the input service filter.

```

[edit]
user@host# edit firewall family inet service-filter in_filter_presvc

```

```

[edit firewall family inet service-filter in_filter_presvc]
user@host# set term t1 from protocol tcp
user@host# set term t1 from source-port bgp
user@host# set term t1 then count svc_in_pkts
user@host# set term t1 then service

```

2. Configure the postservice input filter.

```

[edit]
user@host# edit firewall family inet service-filter in_filter_postsvc

```

```

[edit firewall family inet service-filter in_filter_postsvc]
user@host# set term t2 from protocol tcp
user@host# set term t2 from source-port bgp
user@host# set term t2 then count svc_in_pkts_rtn
user@host# set term t2 then skip

```

3. Configure the output service filter.

```

[edit]
user@host# edit firewall family inet service-filter out_filter_presvc

```

```

[edit firewall family inet service-filter out_filter_presvc]
user@host# set term t3 from protocol icmp
user@host# set term t3 from destination-port bgp
user@host# set term t3 then count svc_out_pkts
user@host# set term t3 then service

```

Results Confirm the configuration of the input and output service filters and the postservice input filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```

[edit]
user@host# show firewall
family inet {
  service-filter in_filter_presvc {
    term t1 {
      from {
        protocol tcp;
        source-port bgp;
      }
    }
  }
}

```

```

        then {
            count svc_in_pkts;
            service;
        }
    }
}
service-filter in_filter_postsvc {
    term t2 {
        from {
            protocol tcp;
            source-port bgp;
        }
        then {
            count svc_in_pkts_rtn;
            skip;
        }
    }
}
service-filter out_filter_presvc {
    term t3 {
        from {
            protocol icmp;
            destination-port bgp;
        }
        then {
            count svc_out_pkts;
            service;
        }
    }
}
}
}

```

Applying the Three Service Filters

Step-by-Step Procedure

To apply the three service filters:

1. Access the IPv4 protocol on the input interface **xe-0/1/0.0**.

```

[edit]
user@host# edit interfaces xe-0/1/0 unit 0 family inet

```

2. Apply the input service filter and the postservice input filter.

```

[edit interfaces xe-0/1/0 unit 0 family inet]
user@host# set service input service-set vrf_svcs service-filter in_filter_presvc
user@host# set service input post-service-filter in_filter_postsvc
user@host# set service output service-set vrf_svcs service-filter out_filter_presvc

```

Results

Confirm the configuration of the interfaces by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```

[edit]
user@host# show interfaces
xe-0/1/0 {
    unit 0 {

```



```

family inet {
  service {
    input {
      service-set vrf_svcs service-filter in_filter_presvc;
      post-service-filter in_filter_postsvc;
    }
    output {
      service-set vrf_svcs service-filter out_filter_presvc;
    }
  }
}
}

```

When you are done configuring the device, commit your candidate configuration.

Verification

Confirm that the configuration is working properly.

- Verifying That Inbound Traffic Is Filtered Before Input Service on page 215
- Verifying That Inbound Traffic Is Filtered After Input Service Processing on page 215
- Verifying That Outbound Traffic Is Filtered Before Output Service Processing on page 215

Verifying That Inbound Traffic Is Filtered Before Input Service

Purpose Verify that inbound packets sent from TCP port 179 are sent for processing by the *input* services associated with the service set **vrf_svcs**.

Action Display the count of packets sent for processing by the *input* services associated with the service set **vrf_svcs**.

```

[edit]
user@host> show firewall filter in_filter_presvc-vrf_svcs counter svc_in_pkts

```

Verifying That Inbound Traffic Is Filtered After Input Service Processing

Purpose Verify that inbound packets sent from TCP port 179 are returned from processing by the *input* services associated with the service set **vrf_svcs**.

Action Display the count of packets returned from processing by the *input* services associated with the service set **vrf_svcs**.

```

[edit]
user@host> show firewall filter in_filter_postsvc-vrf_svcs counter svc_in_pkts_rtn

```

Verifying That Outbound Traffic Is Filtered Before Output Service Processing

Purpose Verify that outbound packets sent to ICMP port 179 are sent for processing by the *output* services associated with the service set **vrf_svcs**.

Action Display the count of packets sent for processing by the *output* services associated with the service set **vrf_svcs**.

[edit]

user@host> **show firewall filter out_filter_presvc-vrf_svcs counter svc_out_pkts**

**Related
Documentation**

- Service Filter Overview on page 61
- How Service Filters Evaluate Packets on page 62
- Guidelines for Configuring Service Filters on page 64
- Guidelines for Applying Service Filters on page 66

CHAPTER 19

Simple Filter Configuration

- Example: Configuring and Applying a Simple Filter on page 217

Example: Configuring and Applying a Simple Filter

This example shows how to configure a simple filter.

- Requirements on page 217
- Overview on page 217
- Configuration on page 218
- Verification on page 220

Requirements

This example uses one of the following hardware components:

- One Gigabit Ethernet intelligent queuing (IQ2) PIC installed on an M120, M320, or T Series router
- One Enhanced Queuing Dense Port Concentrator (EQ DPC) installed on an MX Series router

Before you begin, make sure that you have:

- Installed your supported router and PIC or DPC and performed the initial router configuration.
- Configured basic Ethernet in the topology, and verified that traffic is flowing in the topology and that ingress IPv4 traffic is flowing into logical interface **ge-0/0/1.0**.

Overview

This simple filter sets the loss priority to low for TCP traffic with source address **1.1.1.1**, sets the loss priority to high for HTTP (Web) traffic with source addresses in the **4.0.0.0/8** range, and sets the loss priority to low for all traffic with destination address **6.6.6.6**.

Topology

The simple filter is applied as an input filter (arriving packets are checking for destination address **6.6.6.6**, not queued output packets) on interface **ge-0/0/1.0**.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see “Using the CLI Editor in Configuration Mode” on page 231.

To configure this example, perform the following tasks:

- Configuring the Simple Firewall Filter on page 218
- Applying the Simple Filter to the Logical Interface Input on page 219

CLI Quick Configuration

To quickly configure this example, copy the following commands into a text file, remove any line breaks, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
set firewall family inet simple-filter sf_classify_1 term 1 from source-address 1.1.1.1/32
set firewall family inet simple-filter sf_classify_1 term 1 from protocol tcp
set firewall family inet simple-filter sf_classify_1 term 1 then loss-priority low
set firewall family inet simple-filter sf_classify_1 term 2 from source-address 4.0.0.0/8
set firewall family inet simple-filter sf_classify_1 term 2 from protocol tcp
set firewall family inet simple-filter sf_classify_1 term 2 from source-port http
set firewall family inet simple-filter sf_classify_1 term 2 then loss-priority high
set firewall family inet simple-filter sf_classify_1 term 3 from destination-address 6.6.6.6/32
set firewall family inet simple-filter sf_classify_1 term 3 then loss-priority low
set firewall family inet simple-filter sf_classify_1 term 3 then forwarding-class best-effort
set interfaces ge-0/0/1 unit 0 family inet simple-filter input sf_classify_1
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.3/30
```

Configuring the Simple Firewall Filter

Step-by-Step Procedure

To configure the simple filter:

1. Create the simple filter **sf_classify_1**.

```
[edit]
user@host# edit firewall family inet simple-filter sf_classify_1
```

2. Configure classification of TCP traffic based on the source IP address.

```
[edit firewall family inet simple-filter sf_classify_1]
user@host# set term 1 from source-address 1.1.1.1/32
user@host# set term 1 from protocol tcp
user@host# set term 1 then loss-priority low
```

3. Configure classification of HTTP traffic based on the source IP address.

```
[edit firewall family inet simple-filter sf_classify_1]
user@host# set term 2 from source-address 4.0.0.0/8
user@host# set term 2 from protocol tcp
user@host# set term 2 from source-port http
user@host# set term 2 then loss-priority high
```

4. Configure classification of other traffic based on the destination IP address.

```
[edit firewall family inet simple-filter sf_classify_1]
user@host# set term 3 from destination-address 6.6.6.6/32
user@host# set term 3 then loss-priority low
```

```
user@host# set term 3 then forwarding-class best-effort
```

Results Confirm the configuration of the simple filter by entering the **show firewall** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show firewall
family inet {
  simple-filter sf_classify_1 {
    term 1 {
      from {
        source-address {
          1.1.1.1/32;
        }
        protocol {
          tcp;
        }
      }
      then loss-priority low;
    }
    term 2 {
      from {
        source-address {
          4.0.0.0/8;
        }
        source-port {
          http;
        }
        protocol {
          tcp;
        }
      }
      then loss-priority high;
    }
    term 3 {
      from {
        destination-address {
          6.6.6.6/32;
        }
      }
      then {
        loss-priority low;
        forwarding-class best-effort;
      }
    }
  }
}
```

Applying the Simple Filter to the Logical Interface Input

Step-by-Step Procedure

To apply the simple filter to the logical interface input:

1. Configure the logical interface to which you will apply the simple filter.

```
[edit]
```

```
user@host# edit interfaces ge-0/0/1 unit 0 family inet
```

2. Configure the interface address for the logical interface.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set address 10.1.2.3/30
```

3. Apply the simple filter to the logical interface input.

```
[edit interfaces ge-0/0/1 unit 0 family inet]
user@host# set simple-filter input sf_classify_1
```

Results Confirm the configuration of the interface by entering the **show interfaces** configuration mode command. If the command output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
ge-0/0/1 {
  unit 0 {
    family inet {
      simple-filter {
        input sf_classify_1;
      }
      address 10.1.2.3/30;
    }
  }
}
```

When you are done configuring the device, commit your candidate configuration.

Verification

Confirm that the configuration is working properly.

- Displaying the Mapping of Forwarding Class Maps and Names to Queue Numbers on page 220
- Displaying CoS Queue Counters for the Interface on page 221
- Displaying CoS Queue Counter Details for the Physical Interface on page 221

[Displaying the Mapping of Forwarding Class Maps and Names to Queue Numbers](#)

Purpose Display the mapping of forwarding class names to queue numbers.

Action Enter the **show class-of-service forwarding-class** operational mode command.

```
[edit]
user@host> show class-of-service forwarding-class
```

For information about the command output, see “**show class-of-service forwarding-class**” in the *Junos OS System Basics and Services Command Reference*.

Displaying CoS Queue Counters for the Interface

Purpose Verify that the class-of-service (CoS) queue counters for the interface reflect the simple filter applied to the logical interface.

Action Enter the **show interfaces** command for the physical interface on which the simple filter is applied, and specify **detail** or **extensive** output level.

[edit]

```
user@host> show interfaces ge-0/0/1 detail
```

In the **Physical interface** section, under **Ingress queues**, the **Queue counters** section displays ingress queue counters for each forwarding class.

For more detailed information about the command output, see “**show interfaces (Gigabit Ethernet)**” or “**show interfaces (10-Gigabit Ethernet)**” in the [Junos OS Interfaces Command Reference](#).

Displaying CoS Queue Counter Details for the Physical Interface

Purpose Verify that the CoS queue counter details for the physical interface reflect the simple filter applied to the logical interface.

Action Enter the **show interfaces queue** command for the physical interface on which the simple filter is applied, and specify the **ingress** option.

[edit]

```
user@host> show interfaces queue ge-0/0/1 ingress
```

For information about the command output, see “**show interfaces queue**” in the [Junos OS Interfaces Command Reference](#).

- Related Documentation**
- Simple Filter Overview on page 69
 - How Simple Filters Evaluate Packets on page 69
 - Guidelines for Configuring Simple Filters on page 70
 - Guidelines for Applying Simple Filters on page 74

CHAPTER 20

Firewall Filter Configuration in Logical Systems

- Example: Configuring a Stateless Firewall Filter to Protect a Logical System Against ICMP Floods on page 223

Example: Configuring a Stateless Firewall Filter to Protect a Logical System Against ICMP Floods

This example shows how to configure a stateless firewall filter that protects against ICMP denial-of-service attacks on a logical system.

- Requirements on page 223
- Overview on page 223
- Configuration on page 224
- Verification on page 226

Requirements

No special configuration beyond device initialization is required before configuring stateless firewall filters.

Overview

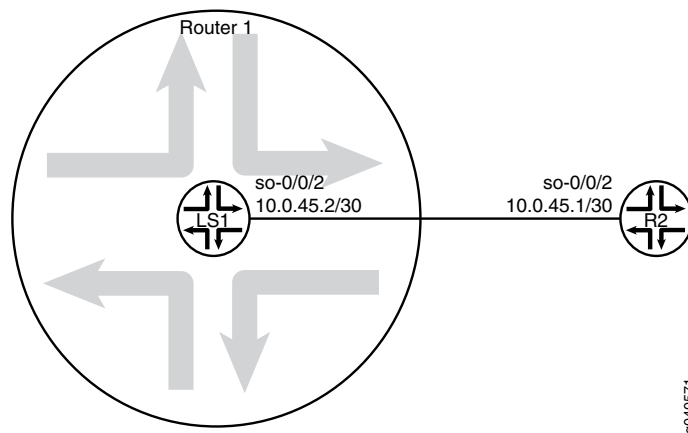
This example shows a stateless firewall filter called **protect-RE** that polices ICMP packets. The **icmp-policer** limits the traffic rate of the ICMP packets to 1,000,000 bps and the burst size to 15,000 bytes. Packets that exceed the traffic rate are discarded.

The policer is incorporated into the action of a filter term called **icmp-term**.

In this example, a ping is sent from a directly connected physical router to the interface configured on the logical system. The logical system accepts the ICMP packets if they are received at a rate of up to 1 Mbps (bandwidth-limit). The logical system drops all ICMP packets when this rate is exceeded. The **burst-size-limit** statement accepts traffic bursts up to 15 Kbps. If bursts exceed this limit, all packets are dropped. When the flow rate subsides, ICMP packets are again accepted.

Figure 3 on page 224 shows the topology used in this example.

Figure 3: Logical System with a Stateless Firewall



Configuration

CLI Quick Configuration To quickly configure an OSPF default route policy on logical systems, copy the following commands into a text file, remove any line breaks, and then paste the commands into the CLI.

```
[edit]
set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet policer input icmp-policer
set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet address 10.0.45.2/30
set logical-systems LS1 firewall family inet filter protect-RE term icmp-term from protocol icmp
set logical-systems LS1 firewall family inet filter protect-RE term icmp-term then policer icmp-policer
set logical-systems LS1 firewall family inet filter protect-RE term icmp-term then accept
set logical-systems LS1 firewall policer icmp-policer if-exceeding bandwidth-limit 1m
set logical-systems LS1 firewall policer icmp-policer if-exceeding burst-size-limit 15k
set logical-systems LS1 firewall policer icmp-policer then discard
```

Step-by-Step Procedure To configure an ICMP firewall filter on a logical system:

1. Configure the interface on the logical system.

```
[edit]
user@host# set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet address 10.0.45.2/30
```

2. Explicitly enable ICMP packets to be received on the interface.

```
[edit]
user@host# set logical-systems LS1 firewall family inet filter protect-RE term icmp-term from protocol icmp
user@host# set logical-systems LS1 firewall family inet filter protect-RE term icmp-term then accept
```

3. Create the policer.

```
[edit]
user@host# set logical-systems LS1 firewall policer icmp-policer if-exceeding bandwidth-limit 1m
```

```
user@host# set logical-systems LS1 firewall policer icmp-policer if-exceeding
burst-size-limit 15k
```

```
user@host# set logical-systems LS1 firewall policer icmp-policer then discard
```

4. Apply the policer to a filter term.

```
[edit]
```

```
user@host# set logical-systems LS1 firewall family inet filter protect-RE term
icmp-term then policer icmp-policer
```

5. Apply the policer to the logical system interface.

```
[edit]
```

```
user@host# set logical-systems LS1 interfaces so-0/0/2 unit 0 family inet policer
input icmp-policer
```

6. If you are done configuring the device, commit the configuration.

```
[edit]
```

```
user@host# commit
```

Results Confirm your configuration by issuing the **show logical-systems LS1** command.

```
user@host# show logical-systems LS1
```

```
interfaces {
  so-0/0/2 {
    unit 0 {
      family inet {
        policer {
          input icmp-policer;
        }
        address 10.0.45.2/30;
      }
    }
  }
}
firewall {
  family inet {
    filter protect-RE {
      term icmp-term {
        from {
          protocol icmp;
        }
        then {
          policer icmp-policer;
          accept;
        }
      }
    }
  }
}
policer icmp-policer {
  if-exceeding {
    bandwidth-limit 1m;
    burst-size-limit 15k;
  }
  then discard;
}
```

```
}
```

Verification

Confirm that the configuration is working properly.

Verifying That Ping Works Unless the Limits Are Exceeded

Purpose Make sure that the logical system interface is protected against ICMP-based DoS attacks.

Action Log in to a system that has connectivity to the logical system and run the **ping** command.

```
user@R2> ping 10.0.45.2
PING 10.0.45.2 (10.0.45.2): 56 data bytes
64 bytes from 10.0.45.2: icmp_seq=0 ttl=64 time=1.316 ms
64 bytes from 10.0.45.2: icmp_seq=1 ttl=64 time=1.277 ms
64 bytes from 10.0.45.2: icmp_seq=2 ttl=64 time=1.269 ms

user@R2> ping 10.0.45.2 size 20000
PING 10.0.45.2 (10.0.45.2): 20000 data bytes
^C
--- 10.0.45.2 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
```

Meaning When you send a normal ping, the packet is accepted. When you send a ping packet that exceeds the filter limit, the packet is discarded.

Related Documentation

- Example: Creating an Interface on a Logical System

PART 3

Administration

- Firewall Filter Standards on page 229
- Firewall Filter Reference on page 231
- Standard Firewall Filter Match Conditions and Actions on page 235
- Service Filter Match Conditions and Actions on page 275
- Reference Information for Firewall Filters in Logical Systems on page 283
- Firewall Filter Statement Hierarchies on page 289
- Summary of Firewall Filter Configuration Statements on page 301

CHAPTER 21

Firewall Filter Standards

- Supported Standards for Filtering on page 229

Supported Standards for Filtering

The Junos OS supports the following RFCs related to filtering:

- RFC 792, *Internet Control Message Protocol*
- RFC 2460, *Internet Protocol, Version 6 (IPv6)*
- RFC 2474, *Definition of the Differentiated Services (DS) Field*
- RFC 2475, *An Architecture for Differentiated Services*
- RFC 2597, *Assured Forwarding PHB Group*
- RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*
- RFC 4291, *IP Version 6 Addressing Architecture*
- RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*

Related Documentation

- Standard Stateless Firewall Filter Overview on page 15
- Service Filter Overview on page 61
- Simple Filter Overview on page 69
- Stateless Firewall Filters in Logical Systems Overview on page 77

CHAPTER 22

Firewall Filter Reference

- Using the CLI Editor in Configuration Mode on page 231

Using the CLI Editor in Configuration Mode

This topic describes some of the basic commands that you must use to enter configuration mode in the command-line interface (CLI) editor, navigate through the configuration hierarchy, get help, and commit or revert the changes that you make during the configuration session. For more information about the commands described here, see the *Junos OS CLI User Guide*.

Task	Command/Statement	Example
Edit Your Configuration		
Enter configuration mode. When you first log in to the device, the device is in operational mode. You must explicitly enter configuration mode. When you do, the CLI prompt changes from user@host> to user@host# and the hierarchy level appears in square brackets.	configure	user@host> configure [edit] user@host#
Create a statement hierarchy. You can use the edit command to simultaneously create a hierarchy and move to that new level in the hierarchy. You cannot use the edit command to change the value of identifiers.	edit <i>hierarchy-level value</i>	[edit] user@host# edit security zones security-zone myzone [edit security zones security-zone myzone] user@host#
Create a statement hierarchy and set identifier values. The set command is similar to edit except that your current level in the hierarchy does not change.	set <i>hierarchy-level value</i>	[edit] user@host# set security zones security-zone myzone [edit] user@host#

Navigate the Hierarchy

Task	Command/Statement	Example
Navigate down to an existing hierarchy level.	<code>edit <i>hierarchy-level</i></code>	[edit] user@host# <code>edit security zones</code> [edit security zones] user@host#
Navigate up one level in the hierarchy.	<code>up</code>	[edit security zones] user@host# <code>up</code> [edit security] user@host#
Navigate to the top of the hierarchy.	<code>top</code>	[edit security zones] user@host# <code>top</code> [edit] user@host#
Commit or Revert Changes		
Commit your configuration.	<code>commit</code>	[edit] user@host# <code>commit</code> commit complete
Roll back changes from the current session. Use the rollback command to revert all changes from the current configuration session. When you run the rollback command before exiting your session or committing changes, the software loads the most recently committed configuration onto the device. You must enter the rollback statement at the edit level in the hierarchy.	<code>rollback</code>	[edit] user@host# <code>rollback</code> load complete
Exit Configuration Mode		
Commit the configuration and exit configuration mode.	<code>commit and-quit</code>	[edit] user@host# <code>commit and-quit</code> user@host>
Exit configuration mode without committing your configuration. You must navigate to the top of the hierarchy using the up or top commands before you can exit configuration mode.	<code>exit</code>	[edit] user@host# <code>exit</code> The configuration has been changed but not committed Exit with uncommitted changes? [yes,no] (yes)
Get Help		

Task	Command/Statement	Example
Display a list of valid options for the current hierarchy level.	?	<pre>[edit] user@host# edit security zones ?</pre> <p>Possible completions:</p> <pre><[Enter]> Execute this command > functional-zone Functional zone > security-zone Security zones Pipe through a command [edit]</pre>

Standard Firewall Filter Match Conditions and Actions

- Standard Firewall Filter Match Conditions for Protocol-Independent Traffic on page 235
- Standard Firewall Filter Match Conditions for IPv4 Traffic on page 236
- Standard Firewall Filter Match Conditions for IPv6 Traffic on page 245
- Standard Firewall Filter Match Conditions for MPLS Traffic on page 250
- Standard Firewall Filter Match Conditions for MPLS-Tagged IPv4 Traffic on page 252
- Standard Firewall Filter Match Conditions for VPLS Traffic on page 254
- Standard Firewall Filter Match Conditions for Layer 2 CCC Traffic on page 261
- Standard Firewall Filter Match Conditions for Layer 2 Bridging Traffic on page 263
- Standard Firewall Filter Terminating Actions on page 269
- Standard Firewall Filter Nonterminating Actions on page 270

Standard Firewall Filter Match Conditions for Protocol-Independent Traffic

You can configure a standard stateless firewall filter with match conditions for protocol-independent traffic (**family any**).



NOTE: Protocol-independent standard firewall filters—firewall filters configured at the [edit firewall family any] hierarchy level— are not supported on the router loopback interface (lo0).

Table 12 on page 236 describes the *match-conditions* you can configure at the [edit firewall family any filter *filter-name* term *term-name* from] hierarchy level.

Table 12: Standard Firewall Filter Match Conditions for Protocol-Independent Traffic

Match Condition	Description
forwarding-class <i>class</i>	Match the forwarding class of the packet. Specify assured-forwarding , best-effort , expedited-forwarding , or network-control . For information about forwarding classes and router-internal output queues, see the <i>Junos OS Class of Service Configuration Guide</i> .
forwarding-class-except <i>class</i>	Do not match on the forwarding class. For details, see the forwarding-class match condition.
interface <i>interface-name</i>	Match the interface on which the packet was received. NOTE: If you configure this match condition with an interface that does not exist, the term does not match any packet.
interface-set <i>interface-set-name</i>	Match the interface on which the packet was received to the specified interface set. To define an interface set, include the interface-set statement at the [edit firewall] hierarchy level. For more information, see “Filtering Packets Received on an Interface Set Overview” on page 53.
packet-length <i>bytes</i>	Match the length of the received packet, in bytes. The length refers only to the IP packet, including the packet header, and does not include any Layer 2 encapsulation overhead.
packet-length-except <i>bytes</i>	Do not match on the received packet length, in bytes. For details, see the packet-length match type.

- Related Documentation**
- Guidelines for Configuring Standard Firewall Filters on page 17
 - Standard Firewall Filter Terminating Actions on page 269
 - Standard Firewall Filter Nonterminating Actions on page 270

Standard Firewall Filter Match Conditions for IPv4 Traffic

You can configure a standard stateless firewall filter with match conditions for Internet Protocol version 4 (IPv4) traffic (**family inet**). Table 13 on page 236 describes the **match-conditions** you can configure at the **[edit firewall family inet filter *filter-name* term *term-name* from]** hierarchy level.

Table 13: Standard Firewall Filter Match Conditions for IPv4 Traffic

Match Condition	Description
address <i>address</i>	Match the IPv4 source or destination address field.
address <i>address</i> except	Do not match the IPv4 source or destination address field.
ah-spi <i>spi-value</i>	(M Series routers, except M120 and M320) Match the IPsec authentication header (AH) security parameter index (SPI) value.

Table 13: Standard Firewall Filter Match Conditions for IPv4 Traffic (*continued*)

Match Condition	Description
ah-spi-except <i>spi-value</i>	(M Series routers, except M120 and M320) Do not match the IPsec AH SPI value.
destination-address <i>address</i>	Match the IPv4 destination address field. You cannot specify both the address and destination-address match conditions in the same term.
destination-address <i>address</i> except	Do not match the IPv4 destination address field. For more information, see the destination-address field.
destination-class <i>class-names</i>	Match one or more specified destination class names (sets of destination prefixes grouped together and given a class name). For more information, see "Firewall Filter Match Conditions Based on Address Classes" on page 39.
destination-class-except <i>class-names</i>	Do not match one or more specified destination class names. For details, see the destination-class match condition.
destination-port <i>number</i>	Match the UDP or TCP destination port field. You cannot specify both the port and destination-port match conditions in the same term. If you configure this match condition, we recommend that you also configure the protocol udp or protocol tcp match statement in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the following text synonyms (the port numbers are also listed): afs (1483), bgp (179), biff (512), bootpc (68), bootps (67), cmd (514), cvspserver (2401), dhcp (67), domain (53), eklogin (2105), ekshell (2106), exec (512), finger (79), ftp (21), ftp-data (20), http (80), https (443), ident (113), imap (143), kerberos-sec (88), klogin (543), kpasswd (761), krb-prop (754), krbupdate (760), kshell (544), ldap (389), ldp (646), login (513), mobileip-agent (434), mobileip-mn (435), msdp (639), netbios-dgm (138), netbios-ns (137), netbios-ssn (139), nfsd (2049), nntp (119), ntalk (518), ntp (123), pop3 (110), pptp (1723), printer (515), radacct (1813), radius (1812), rip (520), rkinit (2108), smtp (25), snmp (161), snmptrap (162), snpp (444), socks (1080), ssh (22), sunrpc (111), syslog (514), tacacs (49), tacacs-ds (65), talk (517), telnet (23), tftp (69), timed (525), who (513), or xmcp (177).
destination-port-except <i>number</i>	Do not match the UDP or TCP destination port field. For details, see the destination-port match condition.
destination-prefix-list <i>name</i>	Match destination prefixes in the specified list. Specify the name of a prefix list defined at the [edit policy-options prefix-list <i>prefix-list-name</i>] hierarchy level.
destination-prefix-list <i>name</i> except	Do not match destination prefixes in the specified list. For more information, see the destination-prefix-list match condition.

Table 13: Standard Firewall Filter Match Conditions for IPv4 Traffic (*continued*)

Match Condition	Description
dscp number	<p>Match the Differentiated Services code point (DSCP). The DiffServ protocol uses the type-of-service (ToS) byte in the IP header. The most significant 6 bits of this byte form the DSCP. For more information, see the Junos OS Class of Service Configuration Guide.</p> <p>You can specify a numeric value from 0 through 63. To specify the value in hexadecimal form, include 0x as a prefix. To specify the value in binary form, include b as a prefix.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed):</p> <ul style="list-style-type: none"> • RFC 3246, <i>An Expedited Forwarding PHB (Per-Hop Behavior)</i>, defines one code point: ef (46). • RFC 2597, <i>Assured Forwarding PHB Group</i>, defines 4 classes, with 3 drop precedences in each class, for a total of 12 code points: <ul style="list-style-type: none"> • af11 (10), af12 (12), af13 (14) • af21 (18), af22 (20), af23 (22) • af31 (26), af32 (28), af33 (30) • af41 (34), af42 (36), af43 (38)
dscp-except number	Do not match on the DSCP number. For more information, see the dscp match condition.
esp-spi spi-value	Match the IPsec encapsulating security payload (ESP) SPI value. Match on this specific SPI value. You can specify the ESP SPI value in hexadecimal, binary, or decimal form.
esp-spi-except spi-value	Match the IPsec ESP SPI value. Do not match on this specific SPI value.
first-fragment	<p>Match if the packet is the first fragment of a fragmented packet. Do not match if the packet is a trailing fragment of a fragmented packet. The first fragment of a fragmented packet has a fragment offset value of 0.</p> <p>This match condition is an alias for the bit-field match condition fragment-offset 0 match condition.</p> <p>To match both first and trailing fragments, you can use two terms that specify different match conditions: first-fragment and is-fragment.</p>
forwarding-class class	<p>Match the forwarding class of the packet.</p> <p>Specify assured-forwarding, best-effort, expedited-forwarding, or network-control.</p> <p>For information about forwarding classes and router-internal output queues, see the Junos OS Class of Service Configuration Guide.</p>
forwarding-class-except class	Do not match the forwarding class of the packet. For details, see the forwarding-class match condition.
fragment-flags number	<p>(Ingress only) Match the three-bit IP fragmentation flags field in the IP header.</p> <p>In place of the numeric field value, you can specify one of the following keywords (the field values are also listed): dont-fragment (0x4), more-fragments (0x2), or reserved (0x8).</p>

Table 13: Standard Firewall Filter Match Conditions for IPv4 Traffic (*continued*)

Match Condition	Description
fragment-offset value	<p>Match the 13-bit fragment offset field in the IP header. The value is the offset, in 8-byte units, in the overall datagram message to the data fragment. Specify a numeric value, a range of values, or a set of values. An offset value of 0 indicates the first fragment of a fragmented packet.</p> <p>The first-fragment match condition is an alias for the fragment-offset 0 match condition.</p> <p>To match both first and trailing fragments, you can use two terms that specify different match conditions (first-fragment and is-fragment).</p>
fragment-offset-except number	Do not match the 13-bit fragment offset field.
icmp-code number	<p>Match the ICMP message code field.</p> <p>If you configure this match condition, we recommend that you also configure the protocol icmp match condition in the same term.</p> <p>If you configure this match condition, you must also configure the icmp-type message-type match condition in the same term. An ICMP message code provides more specific information than an ICMP message type, but the meaning of an ICMP message code is dependent on the associated ICMP message type.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed). The keywords are grouped by the ICMP type with which they are associated:</p> <ul style="list-style-type: none"> parameter-problem: ip-header-bad (0), required-option-missing (1) redirect: redirect-for-host (1), redirect-for-network (0), redirect-for-tos-and-host (3), redirect-for-tos-and-net (2) time-exceeded: ttl-eq-zero-during-reassembly (1), ttl-eq-zero-during-transit (0) unreachable: communication-prohibited-by-filtering (13), destination-host-prohibited (10), destination-host-unknown (7), destination-network-prohibited (9), destination-network-unknown (6), fragmentation-needed (4), host-precedence-violation (14), host-unreachable (1), host-unreachable-for-TOS (12), network-unreachable (0), network-unreachable-for-TOS (11), port-unreachable (3), precedence-cutoff-in-effect (15), protocol-unreachable (2), source-host-isolated (8), source-route-failed (5)
icmp-code-except message-code	Do not match the ICMP message code field. For details, see the icmp-code match condition.
icmp-type number	<p>Match the ICMP message type field.</p> <p>If you configure this match condition, we recommend that you also configure the protocol icmp match condition in the same term.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): echo-reply (0), echo-request (8), info-reply (16), info-request (15), mask-request (17), mask-reply (18), parameter-problem (12), redirect (5), router-advertisement (9), router-solicit (10), source-quench (4), time-exceeded (11), timestamp (13), timestamp-reply (14), or unreachable (3).</p>
icmp-type-except message-type	Do not match the ICMP message type field. For details, see the icmp-type match condition.

Table 13: Standard Firewall Filter Match Conditions for IPv4 Traffic (*continued*)

Match Condition	Description
interface <i>interface-name</i>	<p>Match the interface on which the packet was received.</p> <p>NOTE: If you configure this match condition with an interface that does not exist, the term does not match any packet.</p>
interface-group <i>group-number</i>	<p>Match the logical interface on which the packet was received to the specified interface group or set of interface groups. For <i>group-number</i>, specify a single value or a range of values from 0 through 255.</p> <p>To assign a logical interface to an interface group <i>group-number</i>, specify the <i>group-number</i> at the [interfaces <i>interface-name</i> unit <i>number</i> family <i>family</i> filter group] hierarchy level.</p> <p>For more information, see “Filtering Packets Received on a Set of Interface Groups Overview” on page 52.</p>
interface-group-except <i>group-number</i>	<p>Do not match the logical interface on which the packet was received to the specified interface group or set of interface groups. For details, see the interface-group match condition.</p>
interface-set <i>interface-set-name</i>	<p>Match the interface on which the packet was received to the specified interface set.</p> <p>To define an interface set, include the interface-set statement at the [edit firewall] hierarchy level. For more information, see “Filtering Packets Received on an Interface Set Overview” on page 53.</p>

Table 13: Standard Firewall Filter Match Conditions for IPv4 Traffic (*continued*)

Match Condition	Description
ip-options values	<p>Match the 8-bit IP option field, if present, to the specified value or list of values.</p> <p>In place of a numeric value, you can specify one of the following text synonyms (the option values are also listed): loose-source-route (131), record-route (7), router-alert (148), security (130), stream-id (136), strict-source-route (137), or timestamp (68).</p> <p>To match <i>any</i> value for the IP option, use the text synonym any. To match on <i>multiple</i> values, specify the list of values within square brackets ('[' and ']'). To match a <i>range</i> of values, use the value specification [<i>value1-value2</i>].</p> <p>For example, the match condition ip-options [0-147] matches on an IP options field that contains the loose-source-route, record-route, or security values, or any other value from 0 through 147. However, this match condition does not match on an IP options field that contains only the router-alert value (148).</p> <p>For most interfaces, a filter term that specifies an ip-option match on one or more <i>specific</i> IP option values (a value other than any) causes packets to be sent to the Routing Engine so that the kernel can parse the IP option field in the packet header.</p> <ul style="list-style-type: none"> For a firewall filter term that specifies an ip-option match on one or more specific IP option values, you cannot specify the count, log, or syslog nonterminating actions <i>unless</i> you also specify the discard terminating action in the same term. This behavior prevents double-counting of packets for a filter applied to a transit interface on the router. Packets processed on the kernel might be dropped in case of a system bottleneck. To ensure that matched packets are instead sent to the Packet Forwarding Engine (where packet processing is implemented in hardware), use the ip-options any match condition. <p>The 10-Gigabit Ethernet Modular Port Concentrator (MPC), 60-Gigabit Ethernet MPC, 60-Gigabit Queuing Ethernet MPC, and 60-Gigabit Ethernet Enhanced Queuing MPC on MX Series routers are capable of parsing the IP option field of the IPv4 packet header. For interfaces configured on those MPCs, <i>all</i> packets that are matched using the ip-options match condition are sent to the Packet Forwarding Engine for processing.</p>
ip-options-except values	Do not match the IP option field to the specified value or list of values. For details about specifying the values , see the ip-options match condition.
is-fragment	<p>Match if the packet is a trailing fragment of a fragmented packet. Do not match the first fragment of a fragmented packet.</p> <p>NOTE: To match both first and trailing fragments, you can use two terms that specify different match conditions (first-fragment and is-fragment).</p>

Table 13: Standard Firewall Filter Match Conditions for IPv4 Traffic (*continued*)

Match Condition	Description
loss-priority level	<p>Match the packet loss priority (PLP) level.</p> <p>Specify a single level or multiple levels: low, medium-low, medium-high, or high.</p> <p>Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers.</p> <p>For IP traffic on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the tri-color statement at the [edit class-of-service] hierarchy level to commit a PLP configuration with any of the four levels specified. If the tri-color statement is not enabled, you can only configure the high and low levels. This applies to all protocol families.</p> <p>For information about the tri-color statement and for information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the Junos OS Class of Service Configuration Guide.</p>
loss-priority-except level	Do not match the PLP level. For details, see the loss-priority match condition.
packet-length bytes	Match the length of the received packet, in bytes. The length refers only to the IP packet, including the packet header, and does not include any Layer 2 encapsulation overhead.
packet-length-except bytes	Do not match the length of the received packet, in bytes. For details, see the packet-length match type.
port number	<p>Match the UDP or TCP source or destination port field.</p> <p>If you configure this match condition, you cannot configure the destination-port match condition or the source-port match condition in the same term.</p> <p>If you configure this match condition, we recommend that you also configure the protocol udp or protocol tcp match statement in the same term to specify which protocol is being used on the port.</p> <p>In place of the numeric value, you can specify one of the text synonyms listed under destination-port.</p>
port-except number	Do not match the UDP or TCP source or destination port field. For details, see the port match condition.
precedence ip-precedence-field	<p>Match the IP precedence field.</p> <p>In place of the numeric field value, you can specify one of the following text synonyms (the field values are also listed): critical-ecp (0xa0), flash (0x60), flash-override (0x80), immediate (0x40), internet-control (0xc0), net-control (0xe0), priority (0x20), or routine (0x00). You can specify precedence in hexadecimal, binary, or decimal form.</p>
prefix-list name	Match the prefixes of the source or destination address fields to the prefixes in the specified list. The prefix list is defined at the [edit policy-options prefix-list prefix-list-name] hierarchy level.
prefix-list name except	Do not match the prefixes of the source or destination address fields to the prefixes in the specified list. For more information, see the prefix-list match condition.

Table 13: Standard Firewall Filter Match Conditions for IPv4 Traffic (*continued*)

Match Condition	Description
protocol <i>number</i>	Match the IP protocol type field. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): ah (51), dstopts (60), egp (8), esp (50), fragment (44), gre (47), hop-by-hop (0), icmp (1), icmp6 (58), icmpv6 (58), igmp (2), ipip (4), ipv6 (41), ospf (89), pim (103), rsvp (46), sctp (132), tcp (6), udp (17), or vrrp (112).
service-filter-hit	Match a packet received from a filter where a service-filter-hit action was applied.
source-address <i>address</i>	Match the IPv4 address of the source node sending the packet. You cannot specify both the address and source-address match conditions in the same term.
source-address <i>address</i> except	Do not match the IPv4 address of the source node sending the packet. For more information, see the source-address match condition.
source-class <i>class-names</i>	Match one or more specified source class names (sets of source prefixes grouped together and given a class name). For more information, see “Firewall Filter Match Conditions Based on Address Classes” on page 39.
source-class-except <i>class-names</i>	Do not match one or more specified source class names. For details, see the source-class match condition.
source-port <i>number</i>	Match the UDP or TCP source port field. You cannot specify the port and source-port match conditions in the same term. If you configure this match condition for IPv4 traffic, we recommend that you also configure the protocol udp or protocol tcp match statement in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the text synonyms listed with the destination-port <i>number</i> match condition.
source-port-except <i>number</i>	Do not match the UDP or TCP source port field. For details, see the source-port match condition.
source-prefix-list <i>name</i>	Match source prefixes in the specified list. Specify the name of a prefix list defined at the [edit policy-options prefix-list <i>prefix-list-name</i>] hierarchy level.
source-prefix-list <i>name</i> except	Do not match source prefixes in the specified list. For more information, see the source-prefix-list match condition.
tcp-established	Match TCP packets of an established TCP session (packets other than the first packet of a connection). This is an alias for tcp-flags "(ack rst)" . This match condition does not implicitly check that the protocol is TCP. To check this, specify the protocol tcp match condition.

Table 13: Standard Firewall Filter Match Conditions for IPv4 Traffic (*continued*)

Match Condition	Description
tcp-flags value	<p>Match one or more of the low-order 6 bits in the 8-bit TCP flags field in the TCP header.</p> <p>To specify individual bit fields, you can specify the following text synonyms or hexadecimal values:</p> <ul style="list-style-type: none"> • fin (0x01) • syn (0x02) • rst (0x04) • push (0x08) • ack (0x10) • urgent (0x20) <p>In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet.</p> <p>You can string together multiple flags using the bit-field logical operators.</p> <p>For combined bit-field match conditions, see the tcp-established and tcp-initial match conditions.</p> <p>If you configure this match condition, we recommend that you also configure the protocol tcp match statement in the same term to specify that the TCP protocol is being used on the port.</p> <p>For IPv4 traffic only, this match condition does not implicitly check whether the datagram contains the first fragment of a fragmented packet. To check for this condition for IPv4 traffic only, use the first-fragment match condition.</p>
tcp-initial	<p>Match the initial packet of a TCP connection. This is an alias for tcp-flags "(lack & syn)".</p> <p>This condition does not implicitly check that the protocol is TCP. If you configure this match condition, we recommend that you also configure the protocol tcp match condition in the same term.</p>
ttl number	<p>Match the IPv4 time-to-live number. Specify a TTL value or a range of TTL values. For number, you can specify one or more values from 0 through 255. This match condition is supported only on M120, M320, MX Series, and T Series routers.</p>
ttl-except number	<p>Do not match on the IPv4 TTL number. For details, see the ttl match condition.</p>
vlan-ether-type value	<p>Match the virtual local area network (VLAN) Ethernet type field of a VPLS packet.</p>
vlan-ether-type-except value	<p>Do not match the VLAN Ethernet type field of a VPLS packet.</p>

**Related
Documentation**

- Guidelines for Configuring Standard Firewall Filters on page 17
- Standard Firewall Filter Terminating Actions on page 269
- Standard Firewall Filter Nonterminating Actions on page 270

Standard Firewall Filter Match Conditions for IPv6 Traffic

You can configure a standard stateless firewall filter with match conditions for Internet Protocol version 6 (IPv6) traffic (**family inet6**). Table 14 on page 245 describes the *match-conditions* you can configure at the [edit firewall family inet6 filter *filter-name* term *term-name* from] hierarchy level.

Table 14: Standard Firewall Filter Match Conditions for IPv6 Traffic

Match Condition	Description
address <i>address</i>	Match the IPv6 source or destination address field.
address <i>address</i> except	Do not match the IPv6 source or destination address field.
destination-address <i>address</i>	Match the IPv6 destination address field. You cannot specify both the address and destination-address match conditions in the same term.
destination-address <i>address</i> except	Do not match the IPv6 destination address field. For more information, see the destination-address field.
destination-class <i>class-names</i>	Match one or more specified destination class names (sets of destination prefixes grouped together and given a class name). For more information, see “Firewall Filter Match Conditions Based on Address Classes” on page 39.
destination-class-except <i>class-names</i>	Do not match one or more specified destination class names. For details, see the destination-class match condition.
destination-port <i>number</i>	Match the UDP or TCP destination port field. You cannot specify both the port and destination-port match conditions in the same term. If you configure this match condition, we recommend that you also configure the next-header udp or next-header tcp match condition in the same term to specify which protocol is being used on the port. In place of the numeric value, you can specify one of the following text synonyms (the port numbers are also listed): afs (1483), bgp (179), biff (512), bootpc (68), bootps (67), cmd (514), cvspserver (2401), dhcp (67), domain (53), eklogin (2105), ekshell (2106), exec (512), finger (79), ftp (21), ftp-data (20), http (80), https (443), ident (113), imap (143), kerberos-sec (88), klogin (543), kpasswd (761), krb-prop (754), krbupdate (760), kshell (544), ldap (389), ldp (646), login (513), mobileip-agent (434), mobileip-mn (435), msdp (639), netbios-dgm (138), netbios-ns (137), netbios-ssn (139), nfsd (2049), nntp (119), ntalk (518), ntp (123), pop3 (110), pptp (1723), printer (515), radacct (1813), radius (1812), rip (520), rkinit (2108), smtp (25), snmp (161), snmptrap (162), snpp (444), socks (1080), ssh (22), sunrpc (111), syslog (514), tacacs (49), tacacs-ds (65), talk (517), telnet (23), tftp (69), timed (525), who (513), or xdmcp (177).
destination-port-except <i>number</i>	Do not match the UDP or TCP destination port field. For details, see the destination-port match condition.
destination-prefix-list <i>prefix-list-name</i>	Match the prefix of the IPv6 destination address field. The prefix list is defined at the [edit policy-options prefix-list <i>prefix-list-name</i>] hierarchy level.

Table 14: Standard Firewall Filter Match Conditions for IPv6 Traffic (*continued*)

Match Condition	Description
destination-prefix-list <i>prefix-list-name</i> except	Do not match the prefix of the IPv6 destination address field. For more information, see the destination-prefix-list match condition.
forwarding-class <i>class</i>	<p>Match the forwarding class of the packet.</p> <p>Specify assured-forwarding, best-effort, expedited-forwarding, or network-control.</p> <p>For information about forwarding classes and router-internal output queues, see the Junos OS Class of Service Configuration Guide.</p>
forwarding-class-except <i>class</i>	Do not match the forwarding class of the packet. For details, see the forwarding-class match condition.
icmp-code <i>message-code</i>	<p>Match the ICMP message code field.</p> <p>If you configure this match condition, we recommend that you also configure the next-header icmp6 or next-header icmpv6 match condition in the same term.</p> <p>If you configure this match condition, you must also configure the icmp-type message-type match condition in the same term. An ICMP message code provides more specific information than an ICMP message type, but the meaning of an ICMP message code is dependent on the associated ICMP message type.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed). The keywords are grouped by the ICMP type with which they are associated:</p> <ul style="list-style-type: none"> parameter-problem: ip6-header-bad (0), unrecognized-next-header (1), unrecognized-option (2) time-exceeded: ttl-eq-zero-during-reassembly (1), ttl-eq-zero-during-transit (0) destination-unreachable: administratively-prohibited (1), address-unreachable (3), no-route-to-destination (0), port-unreachable (4)
icmp-code-except <i>message-code</i>	Do not match the ICMP message code field. For details, see the icmp-code match condition.
icmp-type <i>message-type</i>	<p>Match the ICMP message type field.</p> <p>If you configure this match condition, we recommend that you also configure the next-header icmp6 or next-header icmpv6 match condition in the same term.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): destination-unreachable (1), echo-reply (129), echo-request (128), membership-query (130), membership-report (131), membership-termination (132), neighbor-advertisement (136), neighbor-solicit (135), node-information-reply (140), node-information-request (139), packet-too-big (2), parameter-problem (4), redirect (137), router-advertisement (134), router-renumbering (138), router-solicit (133), or time-exceeded (3).</p>
icmp-type-except <i>message-type</i>	Do not match the ICMP message type field. For details, see the icmp-type match condition.

Table 14: Standard Firewall Filter Match Conditions for IPv6 Traffic (*continued*)

Match Condition	Description
interface <i>interface-name</i>	<p>Match the interface on which the packet was received.</p> <p>NOTE: If you configure this match condition with an interface that does not exist, the term does not match any packet.</p>
interface-group group-number	<p>Match the logical interface on which the packet was received to the specified interface group or set of interface groups. For group-number, specify a single value or a range of values from 0 through 255.</p> <p>To assign a logical interface to an interface group group-number, specify the group-number at the [interfaces interface-name unit number family family filter group] hierarchy level.</p> <p>For more information, see “Filtering Packets Received on a Set of Interface Groups Overview” on page 52.</p>
interface-group-except group-number	Do not match the logical interface on which the packet was received to the specified interface group or set of interface groups. For details, see the interface-group match condition.
interface-set interface-set-name	<p>Match the interface on which the packet was received to the specified interface set.</p> <p>To define an interface set, include the interface-set statement at the [edit firewall] hierarchy level. For more information, see “Filtering Packets Received on an Interface Set Overview” on page 53.</p>
loss-priority level	<p>Match the packet loss priority (PLP) level.</p> <p>Specify a single level or multiple levels: low, medium-low, medium-high, or high.</p> <p>Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers.</p> <p>For IP traffic on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the tri-color statement at the [edit class-of-service] hierarchy level to commit a PLP configuration with any of the four levels specified. If the tri-color statement is not enabled, you can only configure the high and low levels. This applies to all protocol families.</p> <p>For information about the tri-color statement and for information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the Junos OS Class of Service Configuration Guide.</p>
loss-priority-except level	Do not match the PLP level. For details, see the loss-priority match condition.
next-header header-type	<p>Match the 8-bit IP protocol field that identifies the type of header immediately following the IPv6 header.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): ah (51), dstops (60), egp (8), esp (50), fragment (44), gre (47), hop-by-hop (0), icmp (1), icmp6 (58), icmpv6 (58), igmp (2), ipip (4), ipv6 (41), no-next-header (59), ospf (89), pim (103), routing (43), rsvp (46), sctp (132), tcp (6), udp (17), or vrp (112).</p>
next-header-except header-type	Do not match the 8-bit IP protocol field that identifies the type of header immediately following the IPv6 header. For details, see the next-header match type.

Table 14: Standard Firewall Filter Match Conditions for IPv6 Traffic (*continued*)

Match Condition	Description
packet-length bytes	Match the length of the received packet, in bytes. The length refers only to the IP packet, including the packet header, and does not include any Layer 2 encapsulation overhead.
packet-length-except bytes	Do not match the length of the received packet, in bytes. For details, see the packet-length match type.
port number	<p>Match the UDP or TCP source or destination port field.</p> <p>If you configure this match condition, you cannot configure the destination-port match condition or the source-port match condition in the same term.</p> <p>If you configure this match condition, we recommend that you also configure the next-header udp or next-header tcp match condition in the same term to specify which protocol is being used on the port.</p> <p>In place of the numeric value, you can specify one of the text synonyms listed under destination-port.</p>
port-except number	Do not match the UDP or TCP source or destination port field. For details, see the port match condition.
prefix-list prefix-list-name	Match the prefixes of the source or destination address fields to the prefixes in the specified list. The prefix list is defined at the [edit policy-options prefix-list prefix-list-name] hierarchy level.
prefix-list prefix-list-name except	Do not match the prefixes of the source or destination address fields to the prefixes in the specified list. For more information, see the prefix-list match condition.
service-filter-hit	Match a packet received from a filter where a service-filter-hit action was applied.
source-address address	<p>Match the IPv6 address of the source node sending the packet.</p> <p>You cannot specify both the address and source-address match conditions in the same term.</p>
source-address address except	Do not match the IPv6 address of the source node sending the packet. For more information, see the source-address match condition.
source-class class-names	Match one or more specified source class names (sets of source prefixes grouped together and given a class name). For more information, see "Firewall Filter Match Conditions Based on Address Classes" on page 39.
source-class-except class-names	Do not match one or more specified source class names. For details, see the source-class match condition.

Table 14: Standard Firewall Filter Match Conditions for IPv6 Traffic (*continued*)

Match Condition	Description
source-port <i>number</i>	<p>Match the UDP or TCP source port field.</p> <p>You cannot specify the port and source-port match conditions in the same term.</p> <p>If you configure this match condition, we recommend that you also configure the next-header udp or next-header tcp match condition in the same term to specify which protocol is being used on the port.</p> <p>In place of the numeric value, you can specify one of the text synonyms listed with the destination-port <i>number</i> match condition.</p>
source-port-except <i>number</i>	Do not match the UDP or TCP source port field. For details, see the source-port match condition.
source-prefix-list <i>name</i>	Match the IPv6 address prefix of the packet source field. Specify a prefix list name defined at the [edit policy-options prefix-list <i>prefix-list-name</i>] hierarchy level.
source-prefix-list <i>name</i> except	Do not match the IPv6 address prefix of the packet source field. For more information, see the source-prefix-list match condition.
tcp-established	<p>Match TCP packets other than the first packet of a connection. This is a text synonym for tcp-flags "(ack rst)" (0x14).</p> <p>NOTE: This condition does not implicitly check that the protocol is TCP. To check this, specify the protocol tcp match condition.</p> <p>If you configure this match condition, we recommend that you also configure the next-header tcp match condition in the same term.</p>
tcp-flags <i>flags</i>	<p>Match one or more of the low-order 6 bits in the 8-bit TCP flags field in the TCP header.</p> <p>To specify individual bit fields, you can specify the following text synonyms or hexadecimal values:</p> <ul style="list-style-type: none"> • fin (0x01) • syn (0x02) • rst (0x04) • push (0x08) • ack (0x10) • urgent (0x20) <p>In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet.</p> <p>You can string together multiple flags using the bit-field logical operators.</p> <p>For combined bit-field match conditions, see the tcp-established and tcp-initial match conditions.</p> <p>If you configure this match condition, we recommend that you also configure the next-header tcp match condition in the same term to specify that the TCP protocol is being used on the port.</p>

Table 14: Standard Firewall Filter Match Conditions for IPv6 Traffic (*continued*)

Match Condition	Description
tcp-initial	<p>Match the initial packet of a TCP connection. This is a text synonym for tcp-flags "(!ack & syn)".</p> <p>This condition does not implicitly check that the protocol is TCP. If you configure this match condition, we recommend that you also configure the next-header tcp match condition in the same term.</p>
traffic-class <i>number</i>	<p>Match the 8-bit field that specifies the class-of-service (CoS) priority of the packet.</p> <p>This field was previously used as the type-of-service (ToS) field in IPv4.</p> <p>You can specify a numeric value from 0 through 63. To specify the value in hexadecimal form, include 0x as a prefix. To specify the value in binary form, include b as a prefix.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed):</p> <ul style="list-style-type: none"> • RFC 3246, <i>An Expedited Forwarding PHB (Per-Hop Behavior)</i>, defines one code point: ef (46). • RFC 2597, <i>Assured Forwarding PHB Group</i>, defines 4 classes, with 3 drop precedences in each class, for a total of 12 code points: <ul style="list-style-type: none"> • af11 (10), af12 (12), af13 (14) • af21 (18), af22 (20), af23 (22) • af31 (26), af32 (28), af33 (30) • af41 (34), af42 (36), af43 (38)
traffic-class-exception <i>number</i>	<p>Do not match the 8-bit field that specifies the CoS priority of the packet. For details, see the traffic-class match description.</p>



NOTE: If you specify an IPv6 address in a match condition (the **address**, **destination-address**, or **source-address** match conditions), use the syntax for text representations described in RFC 2373, *IP Version 6 Addressing Architecture*. For more information about IPv6 addresses, see “IPv6 Overview” and “IPv6 Standards” in the *Junos OS Routing Protocols Configuration Guide*.

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 17
- Standard Firewall Filter Terminating Actions on page 269
- Standard Firewall Filter Nonterminating Actions on page 270

Standard Firewall Filter Match Conditions for MPLS Traffic

You can configure a standard stateless firewall filter with match conditions for MPLS traffic (**family mpls**).



NOTE: The input-list *filter-names* and output-list *filter-names* statements for firewall filters for the mpls protocol family are supported on all interfaces with the exception of management interfaces and internal Ethernet interfaces (fxp or em0), loopback interfaces (lo0), and USB modem interfaces (umd).

Table 15 on page 251 describes the *match-conditions* you can configure at the [edit firewall family mpls filter *filter-name* term *term-name* from] hierarchy level.

Table 15: Standard Firewall Filter Match Conditions for MPLS Traffic

Match Condition	Description
exp number	Experimental (EXP) bit number or range of bit numbers in the MPLS header. For <i>number</i> , you can specify one or more values from 0 through 7 in decimal, binary, or hexadecimal format.
exp-except number	Do not match on the EXP bit number or range of bit numbers in the MPLS header. For <i>number</i> , you can specify one or more values from 0 through 7.
forwarding-class class	Forwarding class. Specify assured-forwarding , best-effort , expedited-forwarding , or network-control .
forwarding-class-except class	Do not match on the forwarding class. Specify assured-forwarding , best-effort , expedited-forwarding , or network-control .
interface interface-name	Interface on which the packet was received. You can configure a match condition that matches packets based on the interface on which they were received. NOTE: If you configure this match condition with an interface that does not exist, the term does not match any packet.
interface-set interface-set-name	Match the interface on which the packet was received to the specified interface set. To define an interface set, include the interface-set statement at the [edit firewall] hierarchy level. For more information, see “Filtering Packets Received on an Interface Set Overview” on page 53.
ip-version number	(Interfaces on Enhanced Scaling flexible PIC concentrators [FPCs] on supported T Series routers only) Inner IP version. To match MPLS-tagged IPv4 packets, match on the text synonym ipv4 .

Table 15: Standard Firewall Filter Match Conditions for MPLS Traffic (*continued*)

Match Condition	Description
loss-priority level	<p>Match the packet loss priority (PLP) level.</p> <p>Specify a single level or multiple levels: low, medium-low, medium-high, or high.</p> <p>Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers.</p> <p>For IP traffic on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the tri-color statement at the [edit class-of-service] hierarchy level to commit a PLP configuration with any of the four levels specified. If the tri-color statement is not enabled, you can only configure the high and low levels. This applies to all protocol families.</p> <p>For information about the tri-color statement and for information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the Junos OS Class of Service Configuration Guide.</p>
loss-priority-except level	Do not match the PLP level. For details, see the loss-priority match condition.

**Related
Documentation**

- Guidelines for Configuring Standard Firewall Filters on page 17
- Standard Firewall Filter Terminating Actions on page 269
- Standard Firewall Filter Nonterminating Actions on page 270

Standard Firewall Filter Match Conditions for MPLS-Tagged IPv4 Traffic

This topic covers the following information:

- Matching on IPv4 Packet Header Address or Port Fields in MPLS Flows on page 252
- IP Address Match Conditions for MPLS Traffic on page 253
- IP Port Match Conditions for MPLS Traffic on page 253

Matching on IPv4 Packet Header Address or Port Fields in MPLS Flows

To support network-based service in a core network, you can configure a standard firewall filter that matches Internet Protocol version 4 (IPv4) packet header fields in MPLS traffic (**family mpls**). The firewall filter can match IPv4 packets as an inner payload of an MPLS packet that has a single MPLS label or up to five MPLS labels stacked together. You can configure match conditions based on IPv4 addresses and IPv4 port numbers in the header.

Standard firewall filters based on MPLS-tagged IPv4 headers are supported for interfaces on Enhanced Scaling flexible PIC concentrators (FPCs) on supported T Series routers only. The feature is not supported for the router loopback interface (**lo0**), the router management interface (**fxp0** or **em0**), or USB modem interfaces (**umd**).

To configure a stateless firewall filter term that matches on address or port fields in the IPv4 header of packets in an MPLS flow, you use the **ip-version ipv4** match condition to specify that the term is to match packets based on inner IP fields:

- To match an MPLS-tagged IPv4 packet on the source or destination address field in the IPv4 header, specify the match condition at the **[edit firewall family mpls filter *filter-name* term *term-name* from ip-version ipv4]** hierarchy level.
- To match an MPLS-tagged IPv4 packet on the source or destination port field in the IPv4 header, specify the match condition at the **[edit firewall family mpls filter *filter-name* term *term-name* from ip-version ipv4 protocol (udp | tcp)]** hierarchy level.

IP Address Match Conditions for MPLS Traffic

Table 16 on page 253 describes the IP address-specific ***match-conditions*** you can configure at the **[edit firewall family mpls filter *filter-name* term *term-name* from ip-version ipv4]** hierarchy level.

Table 16: IP Address-Specific Firewall Filter Match Conditions for MPLS Traffic

Match Condition	Description
destination-address <i>address</i>	Match the 32-bit IPv4 address of the destination node to receive the packet.
destination-address <i>address</i> except	Do not match the 32-bit IPv4 address of the destination node to receive the packet.
protocol <i>number</i>	Match the IP protocol type field. In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): ah (51), dstopts (60), egp (8), esp (50), fragment (44), gre (47), hop-by-hop (0), icmp (1), icmp6 (58), icmpv6 (58), igmp (2), ipip (4), ipv6 (41), ospf (89), pim (103), rsvp (46), sctp (132), tcp (6), udp (17), or vrrp (112).
source-address <i>address</i>	Match the 32-bit IPv4 address of the source node sending the packet.
source-address <i>address</i> except	Do not match the 32-bit IPv4 address of the source node sending the packet.

IP Port Match Conditions for MPLS Traffic

Table 17 on page 254 describes the IP port-specific ***match-conditions*** you can configure at the **[edit firewall family mpls filter *filter-name* term *term-name* from ip-version ipv4 protocol (udp | tcp)]** hierarchy level.

Table 17: IP Port-Specific Firewall Filter Match Conditions for MPLS Traffic

Match Condition	Description
destination-port <i>number</i>	<p>Match on the UDP or TCP destination port field.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the port numbers are also listed): afs (1483), bgp (179), biff (512), bootpc (68), bootps (67), cmd (514), cvspserver (2401), dhcp (67), domain (53), eklogin (2105), ekshell (2106), exec (512), finger (79), ftp (21), ftp-data (20), http (80), https (443), ident (113), imap (143), kerberos-sec (88), klogin (543), kpasswd (761), krb-prop (754), krbupdate (760), kshell (544), ldap (389), ldp (646), login (513), mobileip-agent (434), mobilip-mn (435), msdp (639), netbios-dgm (138), netbios-ns (137), netbios-ssn (139), nfsd (2049), nnntp (119), ntalk (518), ntp (123), pop3 (110), pptp (1723), printer (515), radacct (1813), radius (1812), rip (520), rkinit (2108), smtp (25), snmp (161), snmptrap (162), snpp (444), socks (1080), ssh (22), sunrpc (111), syslog (514), tacacs (49), tacacs-ds (65), talk (517), telnet (23), tftp (69), timed (525), who (513), or xmcp (177).</p>
destination-port-except <i>number</i>	<p>Do not match on the UDP or TCP destination port field.</p> <p>In place of the numeric value, you can specify one of the text synonyms listed with the destination-port match condition.</p>
source-port <i>number</i>	<p>Match on the TCP or UDP source port field.</p> <p>In place of the numeric field, you can specify one of the text synonyms listed under destination-port.</p>
source-port-except <i>number</i>	Do not match on the TCP or UDP source port field.

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 17
- Standard Firewall Filter Terminating Actions on page 269
- Standard Firewall Filter Nonterminating Actions on page 270

Standard Firewall Filter Match Conditions for VPLS Traffic

In the **from** statement in the VPLS filter term, you specify conditions that the packet must match for the action in the **then** statement to be taken. All conditions in the **from** statement must match for the action to be taken. The order in which you specify match conditions is not important, because a packet must match all the conditions in a term for a match to occur.

If you specify no match conditions in a term, that term matches all packets.

An individual condition in a **from** statement can contain a list of values. For example, you can specify numeric ranges or multiple source or destination addresses. When a condition defines a list of values, a match occurs if one of the values in the list matches the packet.

Individual conditions in a **from** statement can be negated. When you negate a condition, you are defining an explicit mismatch. For example, the negated match condition for **forwarding-class** is **forwarding-class-except**. If a packet matches a negated condition, it is immediately considered not to match the **from** statement, and the next term in the filter is evaluated, if there is one. If there are no more terms, the packet is discarded.

You can configure a standard firewall filter with match conditions for Virtual Private LAN Service (VPLS) traffic (**family vpls**). Table 18 on page 255 describes the **match-conditions** you can configure at the **[edit firewall family vpls filter *filter-name* term *term-name* from]** hierarchy level.



NOTE: Not all match conditions for VPLS traffic are supported on all routing platforms. A number of match conditions for VPLS traffic are supported only on MX Series 3D Universal Edge Routers.

Table 18: Standard Firewall Filter Match Conditions for VPLS Traffic

Match Condition	Description
destination-mac-address address	Match the destination media access control (MAC) address of a VPLS packet.
destination-port number	<p>(MX Series routers only) Match the UDP or TCP destination port field.</p> <p>You cannot specify both the port and destination-port match conditions in the same term.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the port numbers are also listed): afs (1483), bgp (179), biff (512), bootpc (68), bootps (67), cmd (514), cvspserver (2401), dhcp (67), domain (53), eklogin (2105), ekshell (2106), exec (512), finger (79), ftp (21), ftp-data (20), http (80), https (443), ident (113), imap (143), kerberos-sec (88), klogin (543), kpasswd (761), krb-prop (754), krbupdate (760), kshell (544), ldap (389), ldp (646), login (513), mobileip-agent (434), mobileip-mn (435), msdp (639), netbios-dgm (138), netbios-ns (137), netbios-ssn (139), nfsd (2049), nntp (119), ntalk (518), ntp (123), pop3 (110), pptp (1723), printer (515), radacct (1813), radius (1812), rip (520), rkinit (2108), smtp (25), snmp (161), snmptrap (162), snpp (444), socks (1080), ssh (22), sunrpc (111), syslog (514), tacacs (49), tacacs-ds (65), talk (517), telnet (23), tftp (69), timed (525), who (513), or xdmcp (177).</p>
destination-port-except number	(MX Series routers only) Do not match on the TCP or UDP destination port field. You cannot specify both the port and destination-port match conditions in the same term.
destination-prefix-list name	<p>(MX Series routers only) Match destination prefixes in the specified list. Specify the name of a prefix list defined at the [edit policy-options prefix-list <i>prefix-list-name</i>] hierarchy level.</p> <p>NOTE: VPLS prefix lists support only IPv4 addresses. IPv6 addresses included in a VPLS prefix list will be discarded.</p>
destination-prefix-list name except	(MX Series routers only) Do not match destination prefixes in the specified list. For more information, see the destination-prefix-list match condition.

Table 18: Standard Firewall Filter Match Conditions for VPLS Traffic (*continued*)

Match Condition	Description
dscp number	<p>(MX Series routers only) Match the Differentiated Services code point (DSCP). The DiffServ protocol uses the type-of-service (ToS) byte in the IP header. The most significant 6 bits of this byte form the DSCP. For more information, see the Junos OS Class of Service Configuration Guide.</p> <p>You can specify a numeric value from 0 through 63. To specify the value in hexadecimal form, include 0x as a prefix. To specify the value in binary form, include b as a prefix.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed):</p> <ul style="list-style-type: none"> • RFC 3246, <i>An Expedited Forwarding PHB (Per-Hop Behavior)</i>, defines one code point: ef (46). • RFC 2597, <i>Assured Forwarding PHB Group</i>, defines 4 classes, with 3 drop precedences in each class, for a total of 12 code points: <p>af11 (10), af12 (12), af13 (14),</p> <p>af21 (18), af22 (20), af23 (22),</p> <p>af31 (26), af32 (28), af33 (30),</p> <p>af41 (34), af42 (36), af43 (38)</p>
dscp-except number	(MX Series routers only) Do not match on the DSCP. For details, see the dscp match condition.
ether-type values	<p>Match the 2-octet IEEE 802.3 Length/EtherType field to the specified value or list of values.</p> <p>You can specify decimal or hexadecimal values from 0 through 65535 (0xFFFF). A value from 0 through 1500 (0x05DC) specifies the length of an Ethernet Version 1 frame. A value from 1536 (0x0600) through 65535 specifies the EtherType (nature of the MAC client protocol) of an Ethernet Version 2 frame.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the hexadecimal values are also listed): aarp (0x80F3), appletalk (0x809B), arp (0x0806), ipv4 (0x0800), ipv6 (0x86DD), mpls-multicast (0x8848), mpls-unicast (0x8847), oam (0x8902), ppp (0x880B), pppoe-discovery (0x8863), pppoe-session (0x8864), sna (0x80D5).</p>
ether-type-except values	<p>Do not match the 2-octet Length/EtherType field to the specified value or list of values.</p> <p>For details about specifying the values, see the ether-type match condition.</p>
forwarding-class class	Match the forwarding class. Specify assured-forwarding , best-effort , expedited-forwarding , or network-control .
forwarding-class-except class	Do not match the forwarding class. For details, see the forwarding-class match condition.

Table 18: Standard Firewall Filter Match Conditions for VPLS Traffic (*continued*)

Match Condition	Description
icmp-code <i>message-code</i>	<p>Match the ICMP message code field.</p> <p>If you configure this match condition, we recommend that you also configure the next-header icmp, next-header icmp6, or next-header icmpv6 match condition in the same term.</p> <p>If you configure this match condition, you must also configure the icmp-type <i>message-type</i> match condition in the same term. An ICMP message code provides more specific information than an ICMP message type, but the meaning of an ICMP message code is dependent on the associated ICMP message type.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed). The keywords are grouped by the ICMP type with which they are associated:</p> <ul style="list-style-type: none"> parameter-problem: ip6-header-bad (0), unrecognized-next-header (1), unrecognized-option (2) time-exceeded: ttl-eq-zero-during-reassembly (1), ttl-eq-zero-during-transit (0) destination-unreachable: address-unreachable (3), administratively-prohibited (1), no-route-to-destination (0), port-unreachable (4)
icmp-code-except <i>message-code</i>	Do not match the ICMP message code field. For details, see the icmp-code match condition.
icmp-code <i>number</i>	<p>(MX Series routers only) Match the ICMP message code field.</p> <p>If you configure this match condition, we recommend that you also configure the ip-protocol icmp, ip-protocol icmp6, or ip-protocol icmpv6 match condition in the same term.</p> <p>If you configure this match condition, you must also configure the icmp-type <i>message-type</i> match condition in the same term. An ICMP message code provides more specific information than an ICMP message type, but the meaning of an ICMP message code is dependent on the associated ICMP message type.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed). The keywords are grouped by the ICMP type with which they are associated:</p> <ul style="list-style-type: none"> parameter-problem: ip6-header-bad (0), unrecognized-next-header (1), unrecognized-option (2) time-exceeded: ttl-eq-zero-during-reassembly (1), ttl-eq-zero-during-transit (0) destination-unreachable: address-unreachable (3), administratively-prohibited (1), no-route-to-destination (0), port-unreachable (4)
icmp-code-except <i>number</i>	(MX Series routers only) Do not match on the ICMP code field. For details, see the icmp-code match condition.

Table 18: Standard Firewall Filter Match Conditions for VPLS Traffic (*continued*)

Match Condition	Description
icmp-type <i>number</i>	<p>(MX Series routers only) Match the ICMP message type field.</p> <p>If you configure this match condition, we recommend that you also configure the ip-protocol icmp, ip-protocol icmp6, or ip-protocol icmpv6 match condition in the same term.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): destination-unreachable (1), echo-reply (129), echo-request (128), membership-query (130), membership-report (131), membership-termination (132), neighbor-advertisement (136), neighbor-solicit (135), node-information-reply (140), node-information-request (139), packet-too-big (2), parameter-problem (4), redirect (137), router-advertisement (134), router-renumbering (138), router-solicit (133), or time-exceeded (3).</p>
icmp-type-except <i>number</i>	<p>(MX Series routers only) Do not match the ICMP message type field. For details, see the icmp-type match condition.</p>
interface <i>interface-name</i>	<p>Interface on which the packet was received. You can configure a match condition that matches packets based on the interface on which they were received.</p> <p>NOTE: If you configure this match condition with an interface that does not exist, the term does not match any packet.</p>
interface-group <i>group-name</i>	<p>Match the logical interface on which the packet was received to the specified interface group or set of interface groups. For group-number, specify a single value or a range of values from 0 through 255.</p> <p>To assign a logical interface to an interface group group-number, specify the group-number at the [interfaces <i>interface-name</i> unit <i>number</i> family <i>family</i> filter <i>group</i>] hierarchy level.</p> <p>For more information, see “Filtering Packets Received on a Set of Interface Groups Overview” on page 52.</p>
interface-group-except <i>group-name</i>	<p>Do not match the logical interface on which the packet was received to the specified interface group or set of interface groups. For details, see the interface-group match condition.</p>
interface-set <i>interface-set-name</i>	<p>Match the interface on which the packet was received to the specified interface set.</p> <p>To define an interface set, include the interface-set statement at the [edit firewall] hierarchy level. For more information, see “Filtering Packets Received on an Interface Set Overview” on page 53.</p>
ip-address <i>address</i>	<p>(MX Series routers only) 32-bit address that supports the standard syntax for IPv4 addresses.</p>
ip-destination-address <i>address</i>	<p>(MX Series routers only) 32-bit address that is the final destination node address for the packet.</p>
ip-precedence <i>ip-precedence-field</i>	<p>(MX Series routers only) IP precedence field. In place of the numeric field value, you can specify one of the following text synonyms (the field values are also listed): critical-ecp (0xa0), flash (0x60), flash-override (0x80), immediate (0x40), internet-control (0xc0), net-control (0xe0), priority (0x20), or routine (0x00).</p>
ip-precedence-except <i>ip-precedence-field</i>	<p>(MX Series routers only) Do not match on the IP precedence field.</p>

Table 18: Standard Firewall Filter Match Conditions for VPLS Traffic (*continued*)

Match Condition	Description
ip-protocol <i>number</i>	(MX Series routers only) IP protocol field.
ip-protocol-except <i>number</i>	(MX Series routers only) Do not match on the IP protocol field.
ip-source-address <i>address</i>	(MX Series routers only) IP address of the source node sending the packet.
learn-vlan-1p-priority <i>number</i>	<p>(MX Series routers only) Match on the IEEE 802.1p learned VLAN priority bits in the provider VLAN tag (the only tag in a single-tag frame with 802.1Q VLAN tags or the outer tag in a dual-tag frame with 802.1Q VLAN tags). Specify a single value or multiple values from 0 through 7.</p> <p>Compare with the user-vlan-1p-priority match condition.</p>
learn-vlan-1p-priority-except <i>number</i>	(MX Series routers only) Do not match on the IEEE 802.1p learned VLAN priority bits. For details, see the learn-vlan-1p-priority match condition.
learn-vlan-id <i>number</i>	(MX Series routers only) VLAN identifier used for MAC learning.
learn-vlan-id-except <i>number</i>	(MX Series routers only) Do not match on the VLAN identifier used for MAC learning.
loss-priority <i>level</i>	<p>Packet loss priority (PLP) level. Specify a single level or multiple levels: low, medium-low, medium-high, or high.</p> <p>Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers.</p> <p>For IP traffic on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the tri-color statement at the [edit class-of-service] hierarchy level to commit a PLP configuration with any of the four levels specified. If the tri-color statement is not enabled, you can only configure the high and low levels. This applies to all protocol families.</p> <p>For information about the tri-color statement and for information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the Junos OS Class of Service Configuration Guide.</p>
loss-priority-except <i>level</i>	<p>Do not match on the packet loss priority level. Specify a single level or multiple levels: low, medium-low, medium-high, or high.</p> <p>For information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the Junos OS Class of Service Configuration Guide.</p>
port <i>number</i>	(MX Series routers only) TCP or UDP source or destination port. You cannot specify both the port match condition and either the destination-port or source-port match condition in the same term.
port-except <i>number</i>	(MX Series routers only) Do not match on the TCP or UDP source or destination port. You cannot specify both the port match condition and either the destination-port or source-port match condition in the same term.

Table 18: Standard Firewall Filter Match Conditions for VPLS Traffic (*continued*)

Match Condition	Description
prefix-list <i>name</i>	<p>(MX Series routers only) Match the destination or source prefixes in the specified list. Specify the name of a prefix list defined at the [edit policy-options prefix-list <i>prefix-list-name</i>] hierarchy level.</p> <p>NOTE: VPLS prefix lists support only IPv4 addresses. IPv6 addresses included in a VPLS prefix list will be discarded.</p>
prefix-list <i>name</i> except	<p>(MX Series routers only) Do not match the destination or source prefixes in the specified list. For more information, see the destination-prefix-list match condition.</p>
source-mac-address <i>address</i>	Source MAC address of a VPLS packet.
source-port <i>number</i>	<p>(MX Series routers only) TCP or UDP source port field. You cannot specify the port and source-port match conditions in the same term.</p>
source-port-except <i>number</i>	<p>(MX Series routers only) Do not match on the TCP or UDP source port field. You cannot specify the port and source-port match conditions in the same term.</p>
source-prefix-list <i>name</i>	<p>(MX Series routers only) Match the source prefixes in the specified prefix list. Specify a prefix list name defined at the [edit policy-options prefix-list <i>prefix-list-name</i>] hierarchy level.</p> <p>NOTE: VPLS prefix lists support only IPv4 addresses. IPv6 addresses included in a VPLS prefix list will be discarded.</p>
source-prefix-list <i>name</i> except	<p>(MX Series routers only) Do not match the source prefixes in the specified prefix list. For more information, see the source-prefix-list match condition.</p>
tcp-flags <i>flags</i>	<p>Match one or more of the low-order 6 bits in the 8-bit TCP flags field in the TCP header.</p> <p>To specify individual bit fields, you can specify the following text synonyms or hexadecimal values:</p> <ul style="list-style-type: none"> • fin (0x01) • syn (0x02) • rst (0x04) • push (0x08) • ack (0x10) • urgent (0x20) <p>In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet.</p> <p>You can string together multiple flags using the bit-field logical operators.</p> <p>If you configure this match condition for IPv6 traffic, we recommend that you also configure the next-header tcp match condition in the same term to specify that the TCP protocol is being used on the port.</p>
traffic-type <i>type-name</i>	<p>(MX Series routers only) Traffic type. Specify broadcast, multicast, unknown-unicast, or known-unicast.</p>

Table 18: Standard Firewall Filter Match Conditions for VPLS Traffic (*continued*)

Match Condition	Description
traffic-type-except <i>type-name</i>	(MX Series routers only) Do not match on the traffic type. Specify broadcast , multicast , unknown-unicast , or known-unicast .
user-vlan-1p-priority <i>number</i>	(MX Series routers only) Match on the IEEE 802.1p user priority bits in the customer VLAN tag (the inner tag in a dual-tag frame with 802.1Q VLAN tags). Specify a single value or multiple values from 0 through 7. Compare with the learn-vlan-1p-priority match condition.
user-vlan-1p-priority-except <i>number</i>	(MX Series routers only) Do not match on the IEEE 802.1p user priority bits. For details, see the user-vlan-1p-priority match condition.
user-vlan-id <i>number</i>	(MX Series routers only) Match the first VLAN identifier that is part of the payload.
user-vlan-id-except <i>number</i>	(MX Series routers only) Do not match on the first VLAN identifier that is part of the payload.
vlan-ether-type <i>value</i>	VLAN Ethernet type field of a VPLS packet.
vlan-ether-type-except <i>value</i>	Do not match on the VLAN Ethernet type field of a VPLS packet.

- Related Documentation**
- Guidelines for Configuring Standard Firewall Filters on page 17
 - Standard Firewall Filter Terminating Actions on page 269
 - Standard Firewall Filter Nonterminating Actions on page 270

Standard Firewall Filter Match Conditions for Layer 2 CCC Traffic

You can configure a standard stateless firewall filter with match conditions for Layer 2 circuit cross-connect (CCC) traffic (**family ccc**).

The following restrictions apply to firewall filters for Layer 2 CCC traffic:

- The **input-list** *filter-names* and **output-list** *filter-names* statements for firewall filters for the **ccc** protocol family are supported on all interfaces with the exception of management interfaces and internal Ethernet interfaces (**fxp** or **em0**), loopback interfaces (**lo0**), and USB modem interfaces (**umd**).
- On MX Series routers only, you cannot apply a Layer 2 CCC stateless firewall filter (a firewall filter configured at the **[edit firewall filter family ccc]** hierarchy level) as an output filter. On MX Series routers, firewall filters configured for the **family ccc** statement can be applied only as input filters.

Table 19 on page 262 describes the *match-conditions* you can configure at the **[edit firewall family ccc filter filter-name term term-name from]** hierarchy level.

Table 19: Standard Firewall Filter Match Conditions for Layer 2 CCC Traffic

Match Condition	Description
destination-mac-address address	<p>(MX Series routers only) Match the destination media access control (MAC) address of a virtual private LAN service (VPLS) packet.</p> <p>To have packets correctly evaluated by this match condition when applied to egress traffic flowing over a CCC circuit from a logical interface on an I-chip DPC in a Layer 2 virtual private network (VPN) routing instance, you must make a configuration change to the Layer 2 VPN routing instance. You must explicitly disable the use of a control word for traffic flowing out over a Layer 2 circuit. The use of a control word is enabled by default for Layer 2 VPN routing instances to support the emulated virtual circuit (VC) encapsulation for Layer 2 circuits.</p> <p>To explicitly disable the use of a control word for Layer 2 VPNs, include the no-control-word statement at either of the following hierarchy levels:</p> <ul style="list-style-type: none"> • [edit routing-instances <i>routing-instance-name</i> protocols l2vpn] • [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols l2vpn] <p>For more information, see “Disabling the Control Word for Layer 2 VPNs” in the Junos OS VPNs Configuration Guide.</p>
forwarding-class class	Forwarding class. Specify assured-forwarding , best-effort , expedited-forwarding , or network-control .
forwarding-class-except class	Do not match on the forwarding class. Specify assured-forwarding , best-effort , expedited-forwarding , or network-control .
interface-group group-number	<p>Match the logical interface on which the packet was received to the specified interface group or set of interface groups. For <i>group-number</i>, specify a single value or a range of values from 0 through 255.</p> <p>To assign a logical interface to an interface group <i>group-number</i>, specify the <i>group-number</i> at the [interfaces <i>interface-name</i> unit <i>number</i> family <i>family</i> filter group] hierarchy level.</p> <p>For more information, see “Filtering Packets Received on a Set of Interface Groups Overview” on page 52.</p>
interface-group-except number	Do not match the logical interface on which the packet was received to the specified interface group or set of interface groups. For details, see the interface-group match condition.
learn-vlan-1p-priority number	<p>(MX Series routers only) Match on the IEEE 802.1p learned VLAN priority bits in the provider VLAN tag (the only tag in a single-tag frame with 802.1Q VLAN tags or the outer tag in a dual-tag frame with 802.1Q VLAN tags). Specify a single value or multiple values from 0 through 7.</p> <p>Compare with the user-vlan-1p-priority match condition.</p>
learn-vlan-1p-priority-except number	(MX Series routers only) Do not match on the IEEE 802.1p learned VLAN priority bits. For details, see the learn-vlan-1p-priority match condition.

Table 19: Standard Firewall Filter Match Conditions for Layer 2 CCC Traffic (*continued*)

Match Condition	Description
loss-priority level	<p>Packet loss priority (PLP) level. Specify a single level or multiple levels: low, medium-low, medium-high, or high.</p> <p>Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers.</p> <p>For IP traffic on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the tri-color statement at the [edit class-of-service] hierarchy level to commit a PLP configuration with any of the four levels specified. If the tri-color statement is not enabled, you can only configure the high and low levels. This applies to all protocol families.</p> <p>For information about the tri-color statement and for information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the Junos OS Class of Service Configuration Guide.</p>
loss-priority-except level	<p>Do not match on the packet loss priority level. Specify a single level or multiple levels: low, medium-low, medium-high, or high.</p> <p>For information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the Junos OS Class of Service Configuration Guide.</p>
user-vlan-1p-priority number	<p>(MX Series routers only) Match on the IEEE 802.1p user priority bits in the customer VLAN tag (the inner tag in a dual-tag frame with 802.1Q VLAN tags). Specify a single value or multiple values from 0 through 7.</p> <p>Compare with the learn-vlan-1p-priority match condition.</p>
user-vlan-1p-priority-except number	<p>(MX Series routers only) Do not match on the IEEE 802.1p user priority bits. For details, see the user-vlan-1p-priority match condition.</p>

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 17
- Standard Firewall Filter Terminating Actions on page 269
- Standard Firewall Filter Nonterminating Actions on page 270

Standard Firewall Filter Match Conditions for Layer 2 Bridging Traffic

On MX Series routers only, you can configure a standard stateless firewall filter with match conditions for Layer 2 bridging traffic (**family bridge**). Table 20 on page 263 describes the **match-conditions** you can configure at the **[edit firewall family bridge filter filter-name term term-name from]** hierarchy level.

Table 20: Standard Firewall Filter Match Conditions for Layer 2 Bridging (MX Series 3D Universal Edge Routers Only)

Match Condition	Description
destination-mac-address address	Destination media access control (MAC) address of a Layer 2 packet in a bridging environment.

Table 20: Standard Firewall Filter Match Conditions for Layer 2 Bridging (MX Series 3D Universal Edge Routers Only) (*continued*)

Match Condition	Description
destination-port <i>number</i>	TCP or UDP destination port field. You cannot specify both the port and destination-port match conditions in the same term.
dscp <i>number</i>	<p>Differentiated Services code point (DSCP). The DiffServ protocol uses the type-of-service (ToS) byte in the IP header. The most significant 6 bits of this byte form the DSCP. For more information, see the Junos OS Class of Service Configuration Guide.</p> <p>You can specify a numeric value from 0 through 63. To specify the value in hexadecimal form, include 0x as a prefix. To specify the value in binary form, include b as a prefix.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed):</p> <ul style="list-style-type: none"> • RFC 3246, <i>An Expedited Forwarding PHB (Per-Hop Behavior)</i>, defines one code point: ef (46). • RFC 2597, <i>Assured Forwarding PHB Group</i>, defines 4 classes, with 3 drop precedences in each class, for a total of 12 code points: <p>af11 (10), af12 (12), af13 (14),</p> <p>af21 (18), af22 (20), af23 (22),</p> <p>af31 (26), af32 (28), af33 (30),</p> <p>af41 (34), af42 (36), af43 (38)</p>
dscp-except <i>number</i>	Do not match on the DSCP number. For more information, see the dscp-except match condition.
ether-type <i>value</i>	<p>Match the 2-octet IEEE 802.3 Length/EtherType field to the specified value or list of values.</p> <p>You can specify decimal or hexadecimal values from 0 through 65535 (0xFFFF). A value from 0 through 1500 (0x05DC) specifies the length of an Ethernet Version 1 frame. A value from 1536 (0x0600) through 65535 specifies the EtherType (nature of the MAC client protocol) of an Ethernet Version 2 frame.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the hexadecimal values are also listed): aarp (0x80F3), appletalk (0x809B), arp (0x0806), ipv4 (0x0800), ipv6 (0x86DD), mpls-multicast (0x8848), mpls-unicast (0x8847), oam (0x8902), ppp (0x880B), pppoe-discovery (0x8863), pppoe-session (0x8864), sna (0x80D5).</p>
ether-type-except <i>value</i>	<p>Do not match the 2-octet IEEE 802.3 Length/EtherType field to the specified value or list of values.</p> <p>For details about specifying the values, see the ether-type match condition.</p>
forwarding class <i>class</i>	Forwarding class. Specify assured-forwarding , best-effort , expedited-forwarding , or network-control .
forwarding-class-except <i>class</i>	Ethernet type field of a Layer 2 packet environment. Specify assured-forwarding , best-effort , expedited-forwarding , or network-control .

Table 20: Standard Firewall Filter Match Conditions for Layer 2 Bridging (MX Series 3D Universal Edge Routers Only) (*continued*)

Match Condition	Description
icmp-code <i>message-code</i>	<p>Match the ICMP message code field.</p> <p>If you configure this match condition, we recommend that you also configure the ip-protocol icmp, ip-protocol icmp6, or ip-protocol icmpv6 match condition in the same term.</p> <p>If you configure this match condition, you must also configure the icmp-type message-type match condition in the same term. An ICMP message code provides more specific information than an ICMP message type, but the meaning of an ICMP message code is dependent on the associated ICMP message type.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed). The keywords are grouped by the ICMP type with which they are associated:</p> <ul style="list-style-type: none"> parameter-problem: ip6-header-bad (0), unrecognized-next-header (1), unrecognized-option (2) time-exceeded: ttl-eq-zero-during-reassembly (1), ttl-eq-zero-during-transit (0) destination-unreachable: address-unreachable (3), administratively-prohibited (1), no-route-to-destination (0), port-unreachable (4)
icmp-code-except <i>message-code</i>	Do not match the ICMP message code field. For details, see the icmp-code match condition.
icmp-type <i>message-type</i>	<p>Match the ICMP message type field.</p> <p>If you configure this match condition, we recommend that you also configure the ip-protocol icmp, ip-protocol icmp6, or ip-protocol icmpv6 match condition in the same term.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): destination-unreachable (1), echo-reply (129), echo-request (128), membership-query (130), membership-report (131), membership-termination (132), neighbor-advertisement (136), neighbor-solicit (135), node-information-reply (140), node-information-request (139), packet-too-big (2), parameter-problem (4), redirect (137), router-advertisement (134), router-renumbering (138), router-solicit (133), or time-exceeded (3).</p>
icmp-type-except <i>message-type</i>	Do not match the ICMP message type field. For details, see the icmp-type match condition.
interface <i>interface-name</i>	<p>Interface on which the packet was received. You can configure a match condition that matches packets based on the interface on which they were received.</p> <p>NOTE: If you configure this match condition with an interface that does not exist, the term does not match any packet.</p>
interface-group <i>group-number</i>	<p>Match the logical interface on which the packet was received to the specified interface group or set of interface groups. For group-number, specify a single value or a range of values from 0 through 255.</p> <p>To assign a logical interface to an interface group group-number, specify the group-number at the [interfaces interface-name unit number family family filter group] hierarchy level.</p> <p>For more information, see “Filtering Packets Received on a Set of Interface Groups Overview” on page 52.</p>

Table 20: Standard Firewall Filter Match Conditions for Layer 2 Bridging (MX Series 3D Universal Edge Routers Only) (*continued*)

Match Condition	Description
interface-group-except <i>number</i>	Do not match the logical interface on which the packet was received to the specified interface group or set of interface groups. For details, see the interface-group match condition.
interface-set <i>interface-set-name</i>	Match the interface on which the packet was received to the specified interface set. To define an interface set, include the interface-set statement at the [edit firewall] hierarchy level. For more information, see “Filtering Packets Received on an Interface Set Overview” on page 53.
ip-address <i>address</i>	32-bit address that supports the standard syntax for IPv4 addresses.
ip-destination-address <i>address</i>	32-bit address that is the final destination node address for the packet.
ip-precedence <i>ip-precedence-field</i>	IP precedence field. In place of the numeric field value, you can specify one of the following text synonyms (the field values are also listed): critical-ecp (0xa0), flash (0x60), flash-override (0x80), immediate (0x40), internet-control (0xc0), net-control (0xe0), priority (0x20), or routine (0x00).
ip-precedence-except <i>ip-precedence-field</i>	Do not match on the IP precedence field.
ip-protocol <i>number</i>	IP protocol field.
ip-source-address <i>address</i>	IP address of the source node sending the packet.
isid <i>number</i>	(Supported with Provider Backbone Bridging [PBB]) Match internet service identifier.
isid-dei <i>number</i>	(Supported with PBB) Match the Internet service identifier drop eligibility indicator (DEI) bit.
isid-dei-except <i>number</i>	(Supported with PBB) Do not match the Internet service identifier DEI bit.
isid-priority-code-point <i>number</i>	(Supported with PBB) Match the Internet service identifier priority code point.
isid-priority-code-point-except <i>number</i>	(Supported with PBB) Do not match the Internet service identifier priority code point.
learn-vlan-1p-priority <i>value</i>	(MX Series routers only) Match on the IEEE 802.1p learned VLAN priority bits in the provider VLAN tag (the only tag in a single-tag frame with 802.1Q VLAN tags or the outer tag in a dual-tag frame with 802.1Q VLAN tags). Specify a single value or multiple values from 0 through 7. Compare with the user-vlan-1p-priority match condition.
learn-vlan-1p-priority-except <i>value</i>	(MX Series routers only) Do not match on the IEEE 802.1p learned VLAN priority bits. For details, see the learn-vlan-1p-priority match condition.
learn-vlan-dei <i>number</i>	(Supported with bridging) Match user virtual LAN (VLAN) identifier DEI bit.

Table 20: Standard Firewall Filter Match Conditions for Layer 2 Bridging (MX Series 3D Universal Edge Routers Only) (*continued*)

Match Condition	Description
learn-vlan-dei-except <i>number</i>	(Supported with bridging) Do not match user VLAN identifier DEI bit.
learn-vlan-id <i>number</i>	VLAN identifier used for MAC learning.
learn-vlan-id-except <i>number</i>	Do not match on the VLAN identifier used for MAC learning.
loss-priority <i>level</i>	<p>Packet loss priority (PLP) level. Specify a single level or multiple levels: low, medium-low, medium-high, or high.</p> <p>Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers.</p> <p>For IP traffic on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the tri-color statement at the [edit class-of-service] hierarchy level to commit a PLP configuration with any of the four levels specified. If the tri-color statement is not enabled, you can only configure the high and low levels. This applies to all protocol families.</p> <p>For information about the tri-color statement and for information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the Junos OS Class of Service Configuration Guide.</p>
loss-priority-except <i>level</i>	<p>Do not match on the packet loss priority level. Specify a single level or multiple levels: low, medium-low, medium-high, or high.</p> <p>For information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the Junos OS Class of Service Configuration Guide.</p>
port <i>number</i>	TCP or UDP source or destination port. You cannot specify both the port match condition and either the destination-port or source-port match conditions in the same term.
source-mac-address <i>address</i>	Source MAC address of a Layer 2 packet.
source-port <i>number</i>	TCP or UDP source port field. You cannot specify the port and source-port match conditions in the same term.

Table 20: Standard Firewall Filter Match Conditions for Layer 2 Bridging (MX Series 3D Universal Edge Routers Only) (*continued*)

Match Condition	Description
tcp-flags flags	<p>Match one or more of the low-order 6 bits in the 8-bit TCP flags field in the TCP header.</p> <p>To specify individual bit fields, you can specify the following text synonyms or hexadecimal values:</p> <ul style="list-style-type: none"> • fin (0x01) • syn (0x02) • rst (0x04) • push (0x08) • ack (0x10) • urgent (0x20) <p>In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet.</p> <p>You can string together multiple flags using the bit-field logical operators.</p> <p>Configuring the tcp-flags match condition requires that you configure the next-header-tcp match condition.</p>
traffic-type type	Traffic type. Specify broadcast , multicast , unknown-unicast , or known-unicast .
traffic-type-except type	Do not match on the traffic type.
user-vlan-1p-priority value	<p>(MX Series routers only) Match on the IEEE 802.1p user priority bits in the customer VLAN tag (the inner tag in a dual-tag frame with 802.1Q VLAN tags). Specify a single value or multiple values from 0 through 7.</p> <p>Compare with the learn-vlan-1p-priority match condition.</p>
user-vlan-1p-priority-except value	(MX Series routers only) Do not match on the IEEE 802.1p user priority bits. For details, see the user-vlan-1p-priority match condition.
user-vlan-id number	(MX Series routers only) Match the first VLAN identifier that is part of the payload.
user-vlan-id-except number	(MX Series routers only) Do not match on the first VLAN identifier that is part of the payload.
vlan-ether-type value	VLAN Ethernet type field of a Layer 2 bridging packet.
vlan-ether-type-except value	Do not match on the VLAN Ethernet type field of a Layer 2 bridging packet.

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 17
- Standard Firewall Filter Terminating Actions on page 269
- Standard Firewall Filter Nonterminating Actions on page 270

Standard Firewall Filter Terminating Actions

Standard stateless firewall filters support different sets of terminating actions for each protocol family.



NOTE: You cannot configure the next term action with a *terminating* action in the same filter term. However, you can configure the next term action with another *nonterminating* action in the same filter term.

Table 21 on page 269 describes the terminating actions you can specify in a standard firewall filter term.

Table 21: Terminating Actions for Standard Firewall Filters

Terminating Action	Description	Protocols
accept	Accept the packet.	<ul style="list-style-type: none"> family any family inet family inet6 family mpls family vpls family ccc family bridge
discard	Discard a packet silently, without sending an Internet Control Message Protocol (ICMP) message. Discarded packets are available for logging and sampling.	<ul style="list-style-type: none"> family any family inet family inet6 family mpls family vpls family ccc family bridge
logical-system <i>logical-system-name</i>	Direct the packet to the specified logical system.	<ul style="list-style-type: none"> family inet family inet6

Table 21: Terminating Actions for Standard Firewall Filters (*continued*)

Terminating Action	Description	Protocols
<code>reject message-type</code>	<p>Reject the packet and return an ICMPv4 or ICMPv6 message:</p> <ul style="list-style-type: none"> If no <i>message-type</i> is specified, a destination unreachable message is returned by default. If tcp-reset is specified as the <i>message-type</i>, tcp-reset is returned only if the packet is a TCP packet. Otherwise, the administratively-prohibited message, which has a value of 13, is returned. If any other <i>message-type</i> is specified, that message is returned. <p>NOTE: Rejected packets can be sampled or logged if you configure the sample or syslog action.</p> <p>The <i>message-type</i> can be one of the following values: address-unreachable, administratively-prohibited, bad-host-tos, bad-network-tos, beyond-scope, fragmentation-needed, host-prohibited, host-unknown, host-unreachable, network-prohibited, network-unknown, network-unreachable, no-route, port-unreachable, precedence-cutoff, precedence-violation, protocol-unreachable, source-host-isolated, source-route-failed, or tcp-reset.</p>	<ul style="list-style-type: none"> family inet family inet6
<code>routing-instance</code> <code>routing-instance-name</code>	Direct the packet to the specified routing instance.	<ul style="list-style-type: none"> family inet family inet6
<code>topology</code> <code>topology-name</code>	Direct the packet to the specified topology.	<ul style="list-style-type: none"> family inet family inet6

Related Documentation

- Guidelines for Configuring Standard Firewall Filters on page 17
- Standard Firewall Filter Nonterminating Actions on page 270

Standard Firewall Filter Nonterminating Actions

Standard stateless firewall filters support different sets of nonterminating actions for each protocol family.



NOTE: You cannot configure the next term action with a *terminating* action in the same filter term. However, you can configure the next term action with another *nonterminating* action in the same filter term.

Table 22 on page 271 describes the nonterminating actions you can configure for a standard firewall filter term.

Table 22: Nonterminating Actions for Standard Firewall Filters

Nonterminating Action	Description	Protocol Families
count <i>counter-name</i>	Count the packet in the named counter.	<ul style="list-style-type: none"> • family any • family inet • family inet6 • family mpls • family vpls • family ccc • family bridge
dscp <i>value</i>	<p>Set the IPv4 Differentiated Services code point (DSCP) bit. You can specify a numerical value from 0 through 63. To specify the value in hexadecimal form, include 0x as a prefix. To specify the value in binary form, include b as a prefix.</p> <p>The default DSCP value is best effort, that is, be or 0.</p> <p>You can also specify on the following text synonyms:</p> <ul style="list-style-type: none"> • af11—Assured forwarding class 1, low drop precedence • af12—Assured forwarding class 1, medium drop precedence • af13—Assured forwarding class 1, high drop precedence • af21—Assured forwarding class 2, low drop precedence • af22—Assured forwarding class 2, medium drop precedence • af23—Assured forwarding class 2, high drop precedence • af31—Assured forwarding class 3, low drop precedence • af32—Assured forwarding class 3, medium drop precedence • af33—Assured forwarding class 3, high drop precedence • af41—Assured forwarding class 4, low drop precedence • af42—Assured forwarding class 4, medium drop precedence • af43—Assured forwarding class 4, high drop precedence • be—Best effort • cs0—Class selector 0 • cs1—Class selector 1 • cs2—Class selector 2 • cs3—Class selector 3 • cs4—Class selector 4 • cs5—Class selector 5 • cs6—Class selector 6 • cs7—Class selector 7 • ef—Expedited forwarding <p>NOTE: The actions dscp 0 or dscp be are supported only on T Series and M320 routers and on the 10-Gigabit Ethernet Modular Port Concentrators (MPC), 60-Gigabit Ethernet MPC, 60-Gigabit Ethernet Queuing MPC, and 60-Gigabit Ethernet Enhanced Queuing MPC on MX Series routers. However, these actions are not supported on Enhanced III Flexible PIC Concentrators (FPCs) on M320 routers.</p>	family inet

Table 22: Nonterminating Actions for Standard Firewall Filters (*continued*)

Nonterminating Action	Description	Protocol Families
forwarding-class <i>class-name</i>	Classify the packet to the named forwarding class: <ul style="list-style-type: none"> • <i>forwarding-class-name</i> • assured-forwarding • best-effort • expedited-forwarding • network-control 	<ul style="list-style-type: none"> • family any • family inet • family inet6 • family mpls • family vpls • family ccc • family bridge
ipsec-sa <i>ipsec-sa</i>	Use the specified IPsec security association. NOTE: This action is not supported on MX Series routers.	family inet
load-balance <i>group-name</i>	Use the specified load-balancing group. NOTE: This action is not supported on MX Series routers.	family inet
log	Log the packet header information in a buffer within the Packet Forwarding Engine. You can access this information by issuing the show firewall log command at the command-line interface (CLI).	<ul style="list-style-type: none"> • family inet • family inet6
loss-priority (high medium-high medium-low low)	<p>Set the packet loss priority (PLP) level.</p> <p>You cannot also configure the three-color-policer nonterminating action for the same firewall filter term. These two nonterminating actions are mutually exclusive.</p> <p>Supported on M120 and M320 routers; M7i and M10i routers with the Enhanced CFEB (CFEB-E); and MX Series routers.</p> <p>For IP traffic on M320, MX Series, and T Series routers with Enhanced II Flexible PIC Concentrators (FPCs), you must include the tri-color statement at the [edit class-of-service] hierarchy level to commit a PLP configuration with any of the four levels specified. If the tri-color statement is not enabled, you can only configure the high and low levels. This applies to all protocol families.</p> <p>For information about the tri-color statement and for information about using behavior aggregate (BA) classifiers to set the PLP level of incoming packets, see the Junos OS Class of Service Configuration Guide.</p>	<ul style="list-style-type: none"> • family any • family inet • family inet6 • family mpls • family vpls • family ccc • family bridge
next-hop-group <i>group-name</i>	Use the specified next-hop group.	family inet
policer <i>policer-name</i>	Name of policer to use to rate-limit traffic.	<ul style="list-style-type: none"> • family any • family inet • family inet6 • family mpls • family vpls • family ccc • family bridge

Table 22: Nonterminating Actions for Standard Firewall Filters (*continued*)

Nonterminating Action	Description	Protocol Families
port-mirror	Port-mirror the packet based on the specified family. Supported on M120 routers, M320 routers configured with Enhanced III FPCs, and MX Series routers only.	<ul style="list-style-type: none"> family inet family inet6 family vpls family ccc family bridge
prefix-action <i>action-name</i>	Count or police packets based on the specified action name.	family inet
sample	<p>Sample the packet.</p> <p>NOTE: The Junos OS does not sample packets originating from the router. If you configure a filter and apply it to the output side of an interface, then only the transit packets going through that interface are sampled. Packets that are sent from the Routing Engine to the Packet Forwarding Engine are not sampled.</p>	<ul style="list-style-type: none"> family inet family inet6 family mpls
service-accounting	<p>Count the packet for service accounting. The count is applied to a specific named counter (<code>_junos-dyn-service-counter</code>) that RADIUS can obtain.</p> <p>For more information, see "Configuring Service Packet Counting" in the <i>Junos OS Subscriber Access Configuration Guide</i>.</p>	<ul style="list-style-type: none"> family inet family inet6
service-filter-hit	<p>(Only if the service-filter-hit flag is marked by a previous filter in the current type of chained filters) Direct the packet to the next type of filters.</p> <p>Indicate to subsequent filters in the chain that the packet was already processed. This action, coupled with the service-filter-hit match condition in receiving filters, helps to streamline filter processing.</p> <p>For more information, see "Configuring Firewall Filter Bypass" in the <i>Junos OS Subscriber Access Configuration Guide</i>.</p>	<ul style="list-style-type: none"> family inet family inet6
syslog	Log the packet to the system log file.	<ul style="list-style-type: none"> family inet family inet6
three-color-policer (single-rate two-rate) <i>policer-name</i>	<p>Police the packet using the specified single-rate or two-rate three-color-policer.</p> <p>You cannot also configure the loss-priority action for the same firewall filter term. These two actions are mutually exclusive.</p>	<ul style="list-style-type: none"> family inet family inet6 family mpls family vpls family ccc family bridge

Table 22: Nonterminating Actions for Standard Firewall Filters (*continued*)

Nonterminating Action	Description	Protocol Families
traffic-class value	<p>Specify the traffic-class code point. You can specify a numerical value from 0 through 63. To specify the value in hexadecimal form, include 0x as a prefix. To specify the value in binary form, include b as a prefix.</p> <p>The default traffic-class value is best effort, that is, be or 0.</p> <p>In place of the numeric value, you can specify one of the following text synonyms:</p> <ul style="list-style-type: none"> • af11—Assured forwarding class 1, low drop precedence • af12—Assured forwarding class 1, medium drop precedence • af13—Assured forwarding class 1, high drop precedence • af21—Assured forwarding class 2, low drop precedence • af22—Assured forwarding class 2, medium drop precedence • af23—Assured forwarding class 2, high drop precedence • af31—Assured forwarding class 3, low drop precedence • af32—Assured forwarding class 3, medium drop precedence • af33—Assured forwarding class 3, high drop precedence • af41—Assured forwarding class 4, low drop precedence • af42—Assured forwarding class 4, medium drop precedence • af43—Assured forwarding class 4, high drop precedence • be—Best effort • cs0—Class selector 0 • cs1—Class selector 1 • cs2—Class selector 2 • cs3—Class selector 3 • cs4—Class selector 4 • cs5—Class selector 5 • cs6—Class selector 6 • cs7—Class selector 7 • ef—Expedited forwarding <p>NOTE: The actions traffic-class 0 or traffic-class be are supported only on T Series and M320 routers and on the 10-Gigabit Ethernet Modular Port Concentrator (MPC), 60-Gigabit Ethernet MPC, 60-Gigabit Ethernet Queuing MPC, and 60-Gigabit Ethernet Enhanced Queuing MPC on MX Series routers. However, these actions are not supported on Enhanced III Flexible PIC Concentrators (FPCs) on M320 routers.</p>	family inet6

- Related Documentation**
- Guidelines for Configuring Standard Firewall Filters on page 17
 - Standard Firewall Filter Terminating Actions on page 269

Service Filter Match Conditions and Actions

- Service Filter Match Conditions for IPv4 or IPv6 Traffic on page 275
- Service Filter Terminating Actions on page 281
- Service Filter Nonterminating Actions on page 282

Service Filter Match Conditions for IPv4 or IPv6 Traffic

Service filters support only a subset of the stateless firewall filter match conditions for IPv4 and IPv6 traffic. Table 23 on page 275 describes the service filter match conditions.

Table 23: Service Filter Match Conditions for IPv4 or IPv6 Traffic

Match Condition	Description	Protocol Families	
address <i>address</i>	Match the IP source or destination address field.	family inet	family inet6
address <i>address</i> except	Do not match the IP source or destination address field.	family inet	family inet6
ah-spi <i>spi-value</i>	(M Series routers, except M120 and M320) Match on the IPsec authentication header (AH) security parameter index (SPI) value.	family inet	—
ah-spi-except <i>spi-value</i>	(M Series routers, except M120 and M320) Do not match on the IPsec AH SPI value.	family inet	—
destination-address <i>address</i>	Match the IP destination address field. You cannot specify both the address and destination-address match conditions in the same term.	family inet	family inet6
destination-address <i>address address</i>	Do not match the IP destination address field. You cannot specify both the address and destination-address match conditions in the same term.	family inet	family inet6

Table 23: Service Filter Match Conditions for IPv4 or IPv6 Traffic (*continued*)

Match Condition	Description	Protocol Families	
destination-port <i>number</i>	<p>Match the UDP or TCP destination port field.</p> <p>You cannot specify both the port and destination-port match conditions in the same term.</p> <p>If you configure this match condition for IPv4 traffic, we recommend that you also configure the protocol udp or protocol tcp match statement in the same term to specify which protocol is being used on the port.</p> <p>If you configure this match condition for IPv6 traffic, we recommend that you also configure the next-header udp or next-header tcp match condition in the same term to specify which protocol is being used on the port.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the port numbers are also listed): afs (1483), bgp (179), biff (512), bootpc (68), bootps (67), cmd (514), cvspserver (2401), dhcp (67), domain (53), eklogin (2105), ekshell (2106), exec (512), finger (79), ftp (21), ftp-data (20), http (80), https (443), ident (113), imap (143), kerberos-sec (88), klogin (543), kpasswd (761), krb-prop (754), krbupdate (760), kshell (544), ldap (389), ldp (646), login (513), mobileip-agent (434), mobileip-mn (435), msdp (639), netbios-dgm (138), netbios-ns (137), netbios-ssn (139), nfsd (2049), nntp (119), ntalk (518), ntp (123), pop3 (110), pptp (1723), printer (515), radacct (1813), radius (1812), rip (520), rinit (2108), smtp (25), snmp (161), snmptrap (162), snpp (444), socks (1080), ssh (22), sunrpc (111), syslog (514), tacacs (49), tacacs-ds (65), talk (517), telnet (23), tftp (69), timed (525), who (513), or xmcp (177).</p>	family inet	family inet6
destination-port-except <i>number</i>	Do not match the UDP or TCP destination port field. For details, see the destination-port match description.	family inet	family inet6
destination-prefix-list <i>name</i>	Match the list of destination prefixes. The prefix list is defined at the [edit policy-options prefix-list <i>prefix-list-name</i>] hierarchy level.	family inet	family inet6
esp-spi <i>value</i>	Match the IPsec encapsulating security payload (ESP) SPI value. Specify a single value or a range of values. You can specify a <i>value</i> in hexadecimal, binary, or decimal form. To specify the value in hexadecimal form, include 0x as a prefix. To specify the value in binary form, include b as a prefix.	family inet	family inet6
esp-spi-except <i>value</i>	Do not match the IPsec ESP SPI value or range of values. For details, see the esp-spi match condition.	family inet	family inet6
first-fragment	<p>Match if the packet is the first fragment of a fragmented packet. Do not match if the packet is a trailing fragment of a fragmented packet. The first fragment of a fragmented packet has a fragment offset value of 0.</p> <p>This match condition is an alias for the bit-field match condition fragment-offset 0 match condition.</p> <p>To match both first and trailing fragments, you can use two terms that specify different match conditions: first-fragment and is-fragment.</p>	family inet	—

Table 23: Service Filter Match Conditions for IPv4 or IPv6 Traffic (*continued*)

Match Condition	Description	Protocol Families	
fragment-flags <i>number</i>	<p>(Ingress only) Match the three-bit IP fragmentation flags field in the IP header.</p> <p>In place of the numeric field value, you can specify one of the following keywords (the field values are also listed): dont-fragment (0x4), more-fragments (0x2), or reserved (0x8).</p>	family inet	—
fragment-offset <i>number</i>	<p>Match the 13-bit fragment offset field in the IP header. The value is the offset, in 8-byte units, in the overall datagram message to the data fragment. Specify a numeric value, a range of values, or a set of values. An offset value of 0 indicates the first fragment of a fragmented packet.</p> <p>The first-fragment match condition is an alias for the fragment-offset 0 match condition.</p> <p>To match both first and trailing fragments, you can use two terms that specify different match conditions (first-fragment and is-fragment).</p>	family inet	—
fragment-offset-except <i>number</i>	Do not match the 13-bit fragment offset field.	family inet	—
interface-group <i>group-number</i>	<p>Match the interface group (set of one or more logical interfaces) on which the packet was received. For group-number, specify a value from 0 through 255.</p> <p>For information about configuring interface groups, see “Filtering Packets Received on a Set of Interface Groups Overview” on page 52.</p>	family inet	family inet6
interface-group-except <i>group-number</i>	Do not match the interface group on which the packet was received. for details, see the interface-group match condition.	family inet	family inet6

Table 23: Service Filter Match Conditions for IPv4 or IPv6 Traffic (*continued*)

Match Condition	Description	Protocol Families
ip-options values	<p>Match the 8-bit IP option field, if present, to the specified value or list of values.</p> <p>In place of a numeric value, you can specify one of the following text synonyms (the option values are also listed): loose-source-route (131), record-route (7), router-alert (148), security (130), stream-id (136), strict-source-route (137), or timestamp (68).</p> <p>To match <i>any</i> value for the IP option, use the text synonym any. To match on <i>multiple</i> values, specify the list of values within square brackets ('[' and ']'). To match a <i>range</i> of values, use the value specification [<i>value1-value2</i>].</p> <p>For example, the match condition ip-options [0-147] matches on an IP options field that contains the loose-source-route, record-route, or security values, or any other value from 0 through 147. However, this match condition does not match on an IP options field that contains only the router-alert value (148).</p> <p>For most interfaces, a filter term that specifies an ip-option match on one or more <i>specific</i> IP option values (a value other than any) causes packets to be sent to the Routing Engine so that the kernel can parse the IP option field in the packet header.</p> <ul style="list-style-type: none"> For a firewall filter term that specifies an ip-option match on one or more specific IP option values, you cannot specify the count, log, or syslog nonterminating actions <i>unless</i> you also specify the discard terminating action in the same term. This behavior prevents double-counting of packets for a filter applied to a transit interface on the router. Packets processed on the kernel might be dropped in case of a system bottleneck. To ensure that matched packets are instead sent to the Packet Forwarding Engine (where packet processing is implemented in hardware), use the ip-options any match condition. <p>The 10-Gigabit Ethernet Modular Port Concentrator (MPC), 60-Gigabit Ethernet MPC, 60-Gigabit Queuing Ethernet MPC, and 60-Gigabit Ethernet Enhanced Queuing MPC on MX Series routers are capable of parsing the IP option field of the IPv4 packet header. For interfaces configured on those MPCs, <i>all</i> packets that are matched using the ip-options match condition are sent to the Packet Forwarding Engine for processing.</p>	family inet —
ip-options-except values	Do not match the IP option field to the specified value or list of values. For details about specifying the values , see the ip-options match condition.	family inet —
is-fragment	<p>Match if the packet is a trailing fragment of a fragmented packet. Do not match the first fragment of a fragmented packet.</p> <p>This match condition is an alias for the bit-field match condition fragment-offset 0 except bits.</p> <p>NOTE: To match both first and trailing fragments, you can use two terms that specify different match conditions (first-fragment and is-fragment).</p>	family inet —

Table 23: Service Filter Match Conditions for IPv4 or IPv6 Traffic (*continued*)

Match Condition	Description	Protocol Families	
port number	<p>Match the UDP or TCP source or destination port field.</p> <p>If you configure this match condition, you cannot configure the destination-port match condition or the source-port match condition in the same term.</p> <p>If you configure this match condition for IPv4 traffic, we recommend that you also configure the protocol udp or protocol tcp match statement in the same term to specify which protocol is being used on the port.</p> <p>If you configure this match condition for IPv6 traffic, we recommend that you also configure the next-header udp or next-header tcp match condition in the same term to specify which protocol is being used on the port.</p> <p>In place of the numeric value, you can specify one of the text synonyms listed under destination-port.</p>	family inet	family inet6
port-except number	Do not match the UDP or TCP source or destination port field. For details, see the port match condition.	family inet	family inet6
prefix-list prefix-list-name	Match the prefixes of the source or destination address fields to the prefixes in the specified list. The prefix list is defined at the [edit policy-options prefix-list prefix-list-name] hierarchy level.	family inet	family inet6
protocol number	<p>Match the IP protocol type field.</p> <p>In place of the numeric value, you can specify one of the following text synonyms (the field values are also listed): ah (51), dstopts (60), egp (8), esp (50), fragment (44), gre (47), hop-by-hop (0), icmp (1), icmp6 (58), icmpv6 (58), igmp (2), ipip (4), ipv6 (41), ospf (89), pim (103), rsvp (46), sctp (132), tcp (6), udp (17), or vrrp (112).</p>	family inet	—
protocol-except number	Do not match the IP protocol type field. For details, see the protocol match condition.	family inet	—
source-address address	<p>Match the IP source address.</p> <p>You cannot specify both the address and source-address match conditions in the same term.</p>	family inet	family inet6
source-address address except	<p>Do not match the IP source address.</p> <p>You cannot specify both the address and source-address match conditions in the same term.</p>	family inet	family inet6

Table 23: Service Filter Match Conditions for IPv4 or IPv6 Traffic (*continued*)

Match Condition	Description	Protocol Families	
source-port <i>number</i>	<p>Match the UDP or TCP source port field.</p> <p>You cannot specify the port and source-port match conditions in the same term.</p> <p>If you configure this match condition for IPv4 traffic, we recommend that you also configure the protocol udp or protocol tcp match statement in the same term to specify which protocol is being used on the port.</p> <p>If you configure this match condition for IPv6 traffic, we recommend that you also configure the next-header udp or next-header tcp match condition in the same term to specify which protocol is being used on the port.</p> <p>In place of the numeric value, you can specify one of the text synonyms listed with the destination-port <i>number</i> match condition.</p>	family inet	family inet6
source-port-except <i>number</i>	Do not match the UDP or TCP source port field. For details, see the source-port match condition.	family inet	family inet6
source-prefix-list <i>name</i>	Match source prefixes in the specified list. Specify the name of a prefix list defined at the [edit policy-options prefix-list <i>prefix-list-name</i>] hierarchy level.	family inet	family inet6
tcp-flags <i>value</i>	<p>Match one or more of the low-order 6 bits in the 8-bit TCP flags field in the TCP header.</p> <p>To specify individual bit fields, you can specify the following text synonyms or hexadecimal values:</p> <ul style="list-style-type: none"> • fin (0x01) • syn (0x02) • rst (0x04) • push (0x08) • ack (0x10) • urgent (0x20) <p>In a TCP session, the SYN flag is set only in the initial packet sent, while the ACK flag is set in all packets sent after the initial packet.</p> <p>You can string together multiple flags using the bit-field logical operators.</p> <p>For combined bit-field match conditions, see the tcp-established and tcp-initial match conditions.</p> <p>If you configure this match condition for IPv4 traffic, we recommend that you also configure the protocol tcp match statement in the same term to specify that the TCP protocol is being used on the port.</p> <p>If you configure this match condition for IPv6 traffic, we recommend that you also configure the next-header tcp match condition in the same term to specify that the TCP protocol is being used on the port.</p>	family inet	family inet6



NOTE: If you specify an IPv6 address in a match condition (the address, destination-address, or source-address match conditions), use the syntax for text representations described in RFC 2373, *IP Version 6 Addressing Architecture*. For more information about IPv6 addresses, see “IPv6 Overview” and “IPv6 Standards” in the *Junos OS Routing Protocols Configuration Guide*.

Related Documentation

- Service Filter Overview on page 61
- Guidelines for Configuring Service Filters on page 64
- Example: Configuring and Applying Service Filters on page 211
- Service Filter Terminating Actions on page 281
- Service Filter Nonterminating Actions on page 282

Service Filter Terminating Actions

Service filters support different sets of terminating actions than standard stateless firewall filters or simple filters.



NOTE: Service filters do not support the next term action.

Table 24 on page 281 describes the terminating actions you can configure in a service filter term.

Table 24: Terminating Actions for Service Filters

Terminating Action	Description	Protocol Families
service	Direct the packet to service processing.	<ul style="list-style-type: none"> • inet • inet6
skip	Let the packet bypass service processing.	<ul style="list-style-type: none"> • inet • inet6

Related Documentation

- Service Filter Overview on page 61
- Guidelines for Configuring Service Filters on page 64
- Example: Configuring and Applying Service Filters on page 211
- Service Filter Match Conditions for IPv4 or IPv6 Traffic on page 275
- Service Filter Nonterminating Actions on page 282

Service Filter Nonterminating Actions

Service filters support different sets of terminating actions for each protocol family.



NOTE: Service filters do not support the **next term** action.

Table 25 on page 282 describes the nonterminating actions you can configure in a service filter term.

Table 25: Nonterminating Actions for Service Filters

Nonterminating Action	Description	Protocol Families
count <i>counter-name</i>	Count the packet in the named counter.	<ul style="list-style-type: none"> • inet • inet6
log	Log the packet header information in a buffer within the Packet Forwarding Engine. You can access this information by issuing the show firewall log command at the command-line interface (CLI).	<ul style="list-style-type: none"> • inet • inet6
port-mirror	Port-mirror the packet based on the specified family. Supported on M120 routers, M320 routers configured with Enhanced III FPCs, and MX Series routers only.	<ul style="list-style-type: none"> • inet • inet6
sample	Sample the packet.	<ul style="list-style-type: none"> • inet • inet6

Related Documentation

- Service Filter Overview on page 61
- Guidelines for Configuring Service Filters on page 64
- Example: Configuring and Applying Service Filters on page 211
- Service Filter Match Conditions for IPv4 or IPv6 Traffic on page 275
- Service Filter Terminating Actions on page 281

Reference Information for Firewall Filters in Logical Systems

- Unsupported Firewall Filter Statements for Logical Systems on page 283
- Unsupported Actions for Firewall Filters in Logical Systems on page 285

Unsupported Firewall Filter Statements for Logical Systems

Table 26 on page 283 shows statements that are supported at the `[edit firewall]` hierarchy level but not at the `[edit logical-systems logical-system-name firewall]` hierarchy level.

Table 26: Unsupported Firewall Statements for Logical Systems

Statement	Example	Description
accounting-profile	<pre>[edit] logical-systems { ls1 { firewall { family inet { filter myfilter { accounting-profile fw-profile; ... term accept-all { then { count counter1; accept; } } } } } } }</pre>	In this example, the <code>accounting-profile</code> statement is not allowed because the accounting profile <code>fw-profile</code> is configured under the <code>[edit accounting-options]</code> hierarchy.

Table 26: Unsupported Firewall Statements for Logical Systems (*continued*)

Statement	Example	Description
hierarchical-policer	<pre>[edit] logical-systems { ls1 { firewall { hierarchical-policer { ... } } } }</pre>	In this example, the hierarchical policer statement requires a class-of-service configuration, which is not supported under logical systems.
load-balance-group	<pre>[edit] logical-systems { ls1 { firewall { load-balance-group lb-group { next-hop-group nh-group; } } } }</pre>	<p>This configuration is not allowed because the next-hop-group nh-group statement must be configured at the [edit forwarding-options next-hop-group] hierarchy level—outside the [edit logical-systems logical-system-name firewall] hierarchy.</p> <p>Currently, the forwarding-options dhcp-relay statement is the only forwarding option supported for logical systems.</p>
virtual-channel	<pre>[edit] logical-systems { ls1 { firewall { family inet { filter foo { term one { from { source-address 10.1.0.0/16; } then { virtual-channel sammy; } } } } } } }</pre>	<p>This configuration is not allowed because the virtual channel sammy refers to an object defined at the [edit class-of-service] hierarchy level, and class of service is not supported for logical systems.</p> <p>NOTE:</p> <p>The virtual-channel statement is supported for J Series devices only, provided the firewall filter is configured outside of a logical-system.</p>

- Related Documentation**
- Stateless Firewall Filters in Logical Systems Overview on page 77
 - Guidelines for Configuring and Applying Firewall Filters in Logical Systems on page 78
 - Unsupported Actions for Firewall Filters in Logical Systems on page 285
 - “Introduction to Logical Systems” in the *Junos OS Logical Systems Configuration Guide*
 - “Logical Systems Operations and Restrictions” in the *Junos OS Logical Systems Configuration Guide*

Unsupported Actions for Firewall Filters in Logical Systems

Table 27 on page 285 describes the firewall filter actions that are supported at the `[edit firewall]` hierarchy level, but not supported at the `[edit logical-systems logical-system-name firewall]` hierarchy level.

Table 27: Unsupported Actions for Firewall Filters in Logical Systems

Firewall Filter Action	Example	Description
Terminating Actions Not Supported in a Logical System		
logical-system	<pre>[edit] logical-systems { ls1 { firewall { family inet { filter foo { term one { from { source-address 10.1.0.0/16; } then { logical-system fred; } } } } } } }</pre>	Because the logical-system action refers to fred —a logical system defined outside the local logical system—, this action is not supported.
Nonterminating Actions Not Supported in a Logical System		
ipsec-sa	<pre>[edit] logical-systems { ls1 { firewall { family inet { filter foo { term one { from { source-address 10.1.0.0/16; } then { ipsec-sa barney; } } } } } } }</pre>	Because the ipsec-sa action modifier references barney —a security association defined outside the local logical system—this action is not supported.

Table 27: Unsupported Actions for Firewall Filters in Logical Systems (*continued*)

Firewall Filter Action	Example	Description
next-hop-group	<pre> [edit] logical-systems { ls1 { firewall { family inet { filter foo { term one { from { source-address 10.1.0.0/16; } then { next-hop-group fred; } } } } } } } </pre>	Because the next-hop-group action refers to fred —an object defined at the [edit forwarding-options next-hop-group] hierarchy level—this action is not supported.
port-mirror	<pre> [edit] logical-systems { ls1 { firewall { family inet { filter foo { term one { from { source-address 10.1.0.0/16; } then { port-mirror; } } } } } } } </pre>	Because the port-mirror action relies on a configuration defined at the [edit forwarding-options port-mirroring] hierarchy level, this action is not supported.

Table 27: Unsupported Actions for Firewall Filters in Logical Systems (*continued*)

Firewall Filter Action	Example	Description
sample	<pre>[edit] logical-systems { ls1 { firewall { family inet { filter foo { term one { from { source-address 10.1.0.0/16; } then { sample; } } } } } } }</pre>	<p>In this example, the sample action depends on the sampling configuration defined under the [edit forwarding-options] hierarchy. Therefore, the sample action is not supported.</p>
syslog	<pre>[edit] logical-systems { ls1 { firewall { family inet { filter icmp-syslog { term icmp-match { from { address { 192.168.207.222/32; } protocol icmp; } then { count packets; syslog; accept; } } term default { then accept; } } } } } }</pre>	<p>In this example, there must be at least one system log (system syslog file filename) with the firewall facility enabled for the icmp-syslog filter's logs to be stored.</p> <p>Because this firewall configuration relies on a configuration outside the logical system, the syslog action modifier is not supported.</p>

Related Documentation

- Stateless Firewall Filters in Logical Systems Overview on page 77
- Guidelines for Configuring and Applying Firewall Filters in Logical Systems on page 78
- Unsupported Firewall Filter Statements for Logical Systems on page 283
- “Introduction to Logical Systems” in the *Junos OS Logical Systems Configuration Guide*

- "Logical Systems Operations and Restrictions" in the *Junos OS Logical Systems Configuration Guide*

Firewall Filter Statement Hierarchies

- Statement Hierarchy for Configuring Interface-Specific Firewall Filters on page 289
- Statement Hierarchy for Applying Interface-Specific Firewall Filters on page 290
- Statement Hierarchy for Assigning Interfaces to Interface Groups on page 291
- Statement Hierarchy for Configuring a Filter to Match on a Set of Interface Groups on page 291
- Statement Hierarchy for Applying Filters to an Interface Group on page 292
- Statement Hierarchy for Defining an Interface Set on page 293
- Statement Hierarchy for Configuring a Filter to Match on an Interface Set on page 293
- Statement Hierarchy for Configuring FBF for IPv4 or IPv6 Traffic on page 294
- Statement Hierarchy for Configuring FBF for MPLS-Tagged IPv4 Traffic on page 295
- Statement Hierarchy for Configuring Routing Instances for FBF on page 297
- Statement Hierarchy for Applying FBF Filters to Interfaces on page 298
- Statement Hierarchy for Configuring Firewall Filter Accounting Profiles on page 299
- Statement Hierarchy for Applying Firewall Filter Accounting Profiles on page 300

Statement Hierarchy for Configuring Interface-Specific Firewall Filters

To enable interface-specific instances for stateless firewall filters, include the **interface-specific** statement in the **filter** *filter-name* or **service-filter** *service-filter-name* stanza. Any counters specified as actions in an interface-specific filter are maintained separately per filter instance. Any policers specified as actions in an interface-specific filter are applied per filter instance.

```
firewall {
  family family-name {
    (filter filter-name | service-filter service-filter-name) {
      ...
      interface-specific;
      ...
      term term-name {
        from {
          match-conditions;
        }
        then {
          count counter-name;
        }
      }
    }
  }
}
```

```
        policer policer-name;  
    }  
}  
...  
}  
}  
]
```

You can include the firewall configuration at one of the following hierarchy levels:

- [\[edit\]](#)
- [\[edit logical-systems *logical-system-name*\]](#)

**Related
Documentation**

- [Interface-Specific Firewall Filter Instances Overview on page 51](#)
- [Statement Hierarchy for Applying Interface-Specific Firewall Filters on page 290](#)
- [Example: Configuring Interface-Specific Firewall Filter Counters on page 180](#)

Statement Hierarchy for Applying Interface-Specific Firewall Filters

To apply an interface-specific stateless firewall filter to a logical interface, include the **input *filter-name*** or **output *filter-name*** statement in the **filter** or **service-filter** stanza of the interfaces configuration:

```
interfaces {  
  interface-name {  
    unit unit-number {  
      family family-name {  
        filter {  
          input filter-name-1;  
          output filter-name-2;  
        }  
        service-filter {  
          input service-filter-name-1;  
          output service-filter-name-2;  
        }  
      }  
    }  
  }  
}
```

You can include the interface configuration at one of the following hierarchy levels:

- [\[edit\]](#)
- [\[edit logical-systems *logical-system-name*\]](#)

**Related
Documentation**

- [Interface-Specific Firewall Filter Instances Overview on page 51](#)
- [Statement Hierarchy for Configuring Interface-Specific Firewall Filters on page 289](#)
- [Example: Configuring Interface-Specific Firewall Filter Counters on page 180](#)

Statement Hierarchy for Assigning Interfaces to Interface Groups

To assign a logical interface to an interface group, specify the group number by including the **group** *interface-group-number* statement in the **filter** stanza:

```
interfaces {
  interface-name {
    unit unit-number {
      family ( inet | inet6 | vpls | ccc | bridge ) {
        filter {
          group interface-group-number;
        }
      }
    }
  }
}
```



NOTE: The number 0 is not a valid number for an interface group.

You can configure the firewall filter at one of the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

Related Documentation

- Filtering Packets Received on a Set of Interface Groups Overview on page 52
- Statement Hierarchy for Configuring a Filter to Match on a Set of Interface Groups on page 291
- Example: Filtering Packets Received on an Interface Group on page 184

Statement Hierarchy for Configuring a Filter to Match on a Set of Interface Groups

You can configure a standard stateless firewall filter or a service filter term that matches packets tagged for a specified interface group or set of interface groups.

To configure a standard stateless firewall filter that matches packets tagged for a specified interface group or set of interface groups, configure a filter term that uses the **interface-group** *interface-group-number* match condition:

```
firewall {
  family (inet | inet6 | vpls | ccc | bridge) {
    filter filter-name {
      term term-name {
        from {
          interface-group interface-group-number;
        }
        then {
          filter-actions;
        }
      }
    }
  }
}
```

```
    }  
  }  
}
```

To configure a service filter that matches packets tagged for a specified interface group or set of interface groups, configure a filter term that uses the **interface-group** *interface-group-name* match condition:

```
firewall {  
  family (inet | inet6) {  
    service-filter filter-name {  
      term term-name {  
        from {  
          interface-group interface-group-number;  
        }  
        then {  
          service-filter-actions;  
        }  
      }  
    }  
  }  
}
```

You can configure the firewall filter at one of the following hierarchy levels:

- [edit]
- [edit logical-systems *logical-system-name*]

Related Documentation

- Filtering Packets Received on a Set of Interface Groups Overview on page 52
- Statement Hierarchy for Assigning Interfaces to Interface Groups on page 291
- Example: Filtering Packets Received on an Interface Group on page 184

Statement Hierarchy for Applying Filters to an Interface Group

To apply a standard stateless firewall filter to an interface group, include the **input** *filter-name* or **output** *filter-name* in the **filter** stanza:

```
interfaces {  
  interface-name {  
    unit unit-number {  
      family family-name {  
        ...  
        filter {  
          input filter-name;  
          output filter-name;  
        }  
      }  
    }  
  }  
}
```

You can include the interface configuration at one of the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

Related Documentation

- Interface-Specific Firewall Filter Instances Overview on page 51
- Statement Hierarchy for Assigning Interfaces to Interface Groups on page 291
- Statement Hierarchy for Configuring a Filter to Match on a Set of Interface Groups on page 291
- Example: Filtering Packets Received on an Interface Group on page 184

Statement Hierarchy for Defining an Interface Set

To configure a named group of interfaces that can be referenced in a stateless firewall filter match condition, use the **interface-set** statement to define the interface-set name and two or more interfaces:

```
firewall {
  interface-set interface-set-name {
    interface-name;
  }
}
```

You can include the statements at one of the following hierarchy levels:

- **[edit firewall]**
- **[edit logical-systems *logical-system-name* firewall]**

To specify that the interface set contains all interfaces of a particular type, you can use the '*' (asterisk) wildcard character. For example, use **fe-*** to specify all Fast Ethernet interfaces.

Related Documentation

- Filtering Packets Received on an Interface Set Overview on page 53
- Statement Hierarchy for Configuring a Filter to Match on an Interface Set on page 293
- Example: Configuring a Rate-Limiting Filter Based on Destination Class on page 167
- Example: Filtering Packets Received on an Interface Set on page 188

Statement Hierarchy for Configuring a Filter to Match on an Interface Set

To configure a standard stateless firewall filter that matches packets tagged for a specified interface group or set of interface groups, configure a filter term that uses the **interface-group** *interface-group-name* match condition:

```
firewall {
  family (any | inet | inet6 | mpls | vpls | bridge) {
    filter filter-name {
      term term-name {
```

```
        from {
            interface-set interface-set-name;
        }
        then {
            filter-actions;
        }
    }
}
```

Related Documentation

- Filtering Packets Received on an Interface Set Overview on page 53
- Statement Hierarchy for Defining an Interface Set on page 293
- Example: Configuring a Rate-Limiting Filter Based on Destination Class on page 167
- Example: Filtering Packets Received on an Interface Set on page 188

Statement Hierarchy for Configuring FBF for IPv4 or IPv6 Traffic

You can configure stateless firewall filters for filter-based forwarding by configuring filter terms that specify the **forwarding-class** *class-name* nonterminating action or the **routing-instance** *routing-instance-name* terminating action:

```
firewall {
    family (inet | inet6) {
        filter filter-name {
            term term-name {
                from {
                    ipv4-or-ipv6-match-conditions;
                }
                then {
                    forwarding-class class-name; #optional
                    other-optional-nonterminating-actions;
                    routing-instance routing-instance-name <topology topology-name>;
                }
            }
        }
    }
}
```

You can include the firewall configuration at one of the following hierarchy levels:

- [edit]
- [edit logical-systems *logical-system-name*]

Related Documentation

- Filter-Based Forwarding Overview on page 53
- Standard Firewall Filter Match Conditions for IPv4 Traffic on page 236
- Standard Firewall Filter Match Conditions for IPv6 Traffic on page 245
- Statement Hierarchy for Configuring Routing Instances for FBF on page 297
- Example: Configuring Filter-Based Forwarding on the Source Address on page 194

Statement Hierarchy for Configuring FBF for MPLS-Tagged IPv4 Traffic

For interfaces on Enhanced Scaling flexible PIC concentrators (FPCs) on supported T Series routers only, you can configure stateless firewall filters for filter-based forwarding with match conditions based on Internet Protocol version 4 (IPv4) packet header fields in an MPLS flow:

- Matching on IPv4 Address Fields on page 295
- Matching on TCP Port Number Fields on page 295
- Matching on UDP Port Number Fields on page 296

Matching on IPv4 Address Fields

To configure a firewall filter term that matches on IP source or destination address fields in the IPv4 header of packets in an MPLS flow, you can specify supported match conditions at the `[edit firewall family mpls filter filter-name term term-name from ip-version ipv4]` hierarchy level:

```
[edit]
firewall {
  family mpls {
    filter filter-name {
      term term-name {
        from {
          ip-protocol ipv4 (
            from {
              address-based-match-conditions;
            }
            then {
              forwarding-class class-name; #optional
              other-optional-nonterminating-actions;
              routing-instance routing-instance-name <topology topology-name>;
            }
          }
        }
        then {
          ...
        }
      }
    }
  }
}
```

Matching on TCP Port Number Fields

To configure a firewall filter term that matches on TCP source or destination port number fields in the IPv4 header of packets in an MPLS flow, you can specify supported match conditions at the `[edit firewall family mpls filter filter-name term term-name from ip-version ipv4 protocol tcp]` hierarchy level:

```
[edit]
firewall {
  family mpls {
    filter filter-name {
      term term-name {
```

```

from {
    ip-protocol ipv4 (
        tcp {
            from {
                tcp-port-based-match-conditions;
            }
            then {
                forwarding-class class-name; #optional
                other-optional-nonterminating-actions;
                routing-instance routing-instance-name <topology topology-name>;
            }
        }
    ]
    udp {
        from {
            udp-port-based-match-conditions;
        }
        then {
            forwarding-class class-name; #optional
            other-optional-nonterminating-actions;
            routing-instance routing-instance-name <topology topology-name>;
        }
    }
}
}
}
then {
    ...
}
}
}
}
}
}
}

```

Matching on UDP Port Number Fields

To configure a firewall filter term that matches on UDP source or destination port number fields in the IPv4 header of packets in an MPLS flow, you can specify supported match conditions at the `[edit firewall family mpls filter filter-name term term-name from ip-version ipv4 protocol udp]` hierarchy level:

```

[edit]
firewall {
    family mpls {
        filter filter-name {
            term term-name {
                from {
                    ip-protocol ipv4 (
                        udp {
                            from {
                                udp-port-based-match-conditions;
                            }
                            then {
                                forwarding-class class-name; #optional
                                other-optional-nonterminating-actions;
                                routing-instance routing-instance-name <topology topology-name>;
                            }
                        ]
                    }
                }
            }
        }
    }
}

```

```

        from {
            udp-port-based-match-conditions;
        }
        then {
            forwarding-class class-name; #optional
            other-optional-nonterminating-actions;
            routing-instance routing-instance-name <topology topology-name>;
        }
    }
}
}
}
then {
    ...
}
}
}
}
}
}
}

```

Related Documentation

- Filter-Based Forwarding Overview on page 53
- Standard Firewall Filter Match Conditions for MPLS-Tagged IPv4 Traffic on page 252
- Statement Hierarchy for Configuring Routing Instances for FBF on page 297
- Example: Configuring Filter-Based Forwarding on the Source Address on page 194

Statement Hierarchy for Configuring Routing Instances for FBF

A routing instance is a collection of routing tables, interfaces, and routing protocol parameters. The set of interfaces belongs to the routing tables, and the routing protocol parameters control the information in the routing tables.

To configure a routing instance for filter-based forwarding:

1. The **instance-type** must be **forwarding**. The **forwarding** routing instance type supports filter-based forwarding, where interfaces are not associated with instances. For this instance type, there is no one-to-one mapping between an interface and a routing instance. All interfaces belong to the default instance **inet.0**.
2. The name of the routing instance name must be the one referenced in the firewall filter action.



NOTE: In Junos OS Release 9.0 and later, you can no longer specify a routing-instance name of **default** or include special characters within the name of a routing instance.

You must also create a routing table group that adds interface routes to the following routing instances:

- Routing instance named in the action

- Default routing table **inet.0**

You create a routing table group to resolve the routes installed in the routing instance to directly connected next hops on that interface. For more information on routing table groups and interface routes, see the [Junos OS Routing Protocols Configuration Guide](#).

```
routing-instances {
  routing-table-name {
    instance-type forwarding;
    routing-options {
      static {
        route destination-prefix nexthop address;
      }
    }
  }
}
```

You can include the **forwarding** routing instance at one of the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

Related Documentation

- Filter-Based Forwarding Overview on page 53
- Statement Hierarchy for Configuring FBF for IPv4 or IPv6 Traffic on page 294
- Statement Hierarchy for Configuring FBF for MPLS-Tagged IPv4 Traffic on page 295
- Example: Configuring Filter-Based Forwarding on the Source Address on page 194

Statement Hierarchy for Applying FBF Filters to Interfaces

To apply filter-based forwarding to a logical interface, include the **input** or **output** statement in the **filter** stanza.



NOTE: An interface configured with filter-based forwarding does not support source-class usage (SCU) filter matching and unicast reverse-path forwarding (RPF) check filters.

```
interfaces {
  interface-name {
    unit unit-number {
      family (inet | inet6 | mpls) {
        filter {
          input filter-name;
          output filter-name;
        }
        address address;
      }
    }
  }
}
```

You can include the interfaces configuration at one of the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

Related Documentation

- Filter-Based Forwarding Overview on page 53
- Statement Hierarchy for Configuring FBF for IPv4 or IPv6 Traffic on page 294
- Statement Hierarchy for Configuring FBF for MPLS-Tagged IPv4 Traffic on page 295
- Statement Hierarchy for Configuring Routing Instances for FBF on page 297
- Example: Configuring Filter-Based Forwarding on the Source Address on page 194

Statement Hierarchy for Configuring Firewall Filter Accounting Profiles

To configure an accounting profile that you can apply to a firewall filter, include the **filter-profile *filter-profile-name*** statement in the **accounting-options** stanza.

```
accounting-options {
  filter-profile filter-profile-name {
    file log-filename {
      archive-sites {
        site-urls;
      }
      files number;
      size bytes;
      start-time time;
      transfer-interval minutes;
    }
    interval minutes;
    counters {
      counter-name-1;
      counter-name-2;
    }
  }
}
```

You can include the accounting options configuration at one of the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

To specify the name of the accounting data log file in the **/var/log** directory to be used in conjunction with the accounting profile, include the **file *log-filename*** statement.

To specify how often statistics are collected for the accounting profile, include the **interval *minutes*** statement.

To specify the names of the firewall filter counters for which filter profile statistics are collected, include the **counters** statement.

- Related Documentation**
- Accounting for Standard Firewall Filters Overview on page 55
 - Statement Hierarchy for Applying Firewall Filter Accounting Profiles on page 300
 - Example: Configuring Statistics Collection for a Standard Firewall Filter on page 201

Statement Hierarchy for Applying Firewall Filter Accounting Profiles

You can apply an accounting profile to a standard stateless firewall filter for any supported protocol family except **family any**.

To apply a filter-accounting profile to a stateless firewall filter, include the **accounting-profile *accounting-profile-name*** statement at the firewall **filter** stanza:

```
firewall {  
  family family-name {  
    filter filter-name {  
      accounting-profile accounting-profile-name;  
      interface-specific;  
      physical-interface-policer;  
      term {  
        filter filter-profile-name;  
      }  
      term term-name {  
        from {  
          match-conditions;  
        }  
        then {  
          actions;  
        }  
      }  
    }  
  }  
}
```

You can include the firewall configuration at one of the following hierarchy levels:

- **[edit]**
- **[edit logical-systems *logical-system-name*]**

- Related Documentation**
- Accounting for Standard Firewall Filters Overview on page 55
 - Statement Hierarchy for Configuring Firewall Filter Accounting Profiles on page 299
 - Example: Configuring Statistics Collection for a Standard Firewall Filter on page 201

Summary of Firewall Filter Configuration Statements

accounting-profile

Syntax	<code>accounting-profile <i>name</i>;</code>
Hierarchy Level	<code>[edit firewall family <i>family-name</i> filter <i>filter-name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Enable collection of accounting data for the specified filter.
Options	<i>name</i> —Name assigned to the accounting profile.
Required Privilege	interface—To view this statement in the configuration.
Level	interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Accounting for Standard Firewall Filters Overview on page 55

family

Syntax	<pre> family <i>family-name</i> { filter <i>filter-name</i> { accounting-profile <i>name</i>; interface-specific; physical-interface-filter; } prefix-action <i>name</i> { count; destination-prefix-length <i>prefix-length</i>; policer <i>policer-name</i>; source-prefix-length <i>prefix-length</i>; subnet-prefix-length <i>prefix-length</i>; } simple-filter <i>filter-name</i> { term <i>term-name</i> { from { <i>match-conditions</i>; } then { <i>action</i>; <i>action-modifiers</i>; } } } } </pre>
Hierarchy Level	[edit firewall], [edit logical-systems <i>logical-system-name</i> firewall]
Release Information	Statement introduced before Junos OS Release 7.4. Logical systems support introduced in Junos OS Release 9.3. simple-filter statement introduced in Junos OS Release 7.6. any family type introduced in Junos OS Release 8.0. bridge family type introduced in Junos OS Release 8.4 (MX Series routers only).
Description	Configure a firewall filter for IP version 4 (IPv4) or IP version 6 (IPv6) traffic. On the MX Series routers only, configure a firewall filter for Layer 2 traffic in a bridging environment.
Options	<p><i>family-name</i>—Version or type of addressing protocol:</p> <ul style="list-style-type: none"> • any—Protocol-independent match conditions. • bridge—(MX Series routers only) Layer 2 packets that are part of bridging domain. • ccc—Layer 2 switching cross-connects. • inet—IPv4 addressing protocol. • inet6—IPv6 addressing protocol. • mpls—MPLS. • vpls—Virtual private LAN service (VPLS).

The remaining statements are explained separately.

Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Guidelines for Configuring Standard Firewall Filters on page 17• Guidelines for Configuring Service Filters on page 64• Guidelines for Configuring Simple Filters on page 70

filter (Applying to a Logical Interface)

Syntax	[edit interfaces <i>interface-name</i> unit <i>logical-unit-number</i> family <i>family</i>], [edit logical-systems <i>logical-system-name</i> interfaces <i>interface-name</i> unit <i>logical-unit-number</i> family <i>family</i>]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Apply a stateless firewall filter to a logical interface at a specific protocol level.
Options	<p>input <i>filter-name</i>—Name of one filter to evaluate when packets are received on the interface.</p> <p>output <i>filter-name</i>—Name of one filter to evaluate when packets are transmitted on the interface.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Guidelines for Configuring Standard Firewall Filters on page 17• Guidelines for Applying Standard Firewall Filters on page 22

filter (Configuring)

Syntax	<pre>filter <i>filter-name</i> { accounting-profile <i>name</i>; interface-specific; physical-interface-filter; term <i>term-name</i> { filter <i>filter-name</i>; from { <i>match-conditions</i>; } then { <i>actions</i>; } } }</pre>
Hierarchy Level	[edit firewall family <i>family-name</i>], [edit logical-systems <i>logical-system-name</i> firewall family <i>family-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Logical systems support introduced in Junos OS Release 9.3. physical-interface-filter statement introduced in Junos OS Release 9.6.
Description	Configure firewall filters.
Options	<p><i>filter-name</i>—Name that identifies the filter. The name can contain letters, numbers, and hyphens (-) and can be up to 64 characters long. To include spaces in the name, enclose it in quotation marks (" ").</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	firewall—To view this statement in the configuration. firewall-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Guidelines for Configuring Standard Firewall Filters on page 17Guidelines for Applying Standard Firewall Filters on page 22

firewall

Syntax	firewall { ... }
Hierarchy Level	[edit], [edit logical-systems <i>logical-system-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Logical systems support introduced in Junos OS Release 9.3.
Description	Configure firewall filters. The statements are explained separately.
Required Privilege Level	firewall—To view this statement in the configuration. firewall-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Guidelines for Configuring Standard Firewall Filters on page 17 Guidelines for Configuring Service Filters on page 64 Guidelines for Configuring Simple Filters on page 70

interface-set

Syntax	interface-set <i>interface-set-name</i> { <i>interface-name</i> ; }
Hierarchy Level	[edit firewall], [edit logical-systems <i>logical-system-name</i> firewall]
Release Information	Statement introduced before Junos OS Release 7.4. Logical systems support introduced in Junos OS Release 9.3.
Description	Configure an interface set.
Options	<i>interface-name</i> —Names of each interface to include in the interface set. You must specify more than one name.
Required Privilege Level	firewall—To view this statement in the configuration. firewall-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Filtering Packets Received on an Interface Set Overview on page 53

interface-specific

Syntax	interface-specific;
Hierarchy Level	[edit firewall family <i>family-name</i> filter <i>filter-name</i>], [edit logical-systems <i>logical-system-name</i> firewall family <i>family-name</i> filter <i>filter-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. Logical systems support introduced in Junos OS Release 9.3.
Description	Configure interface-specific names for firewall counters.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Interface-Specific Firewall Filter Instances Overview on page 51

prefix-list

Syntax	<pre>prefix-list name { ip-addresses; apply-path path; }</pre>
Hierarchy Level	[edit dynamic policy-options], [edit logical-systems <i>logical-system-name</i> policy-options], [edit policy-options]
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Support for configuration in the dynamic database introduced in Junos OS Release 9.5.</p> <p>Support for configuration in the dynamic database introduced in Junos OS Release 9.5 for EX Series switches.</p> <p>Support for the vpls protocol family introduced in Junos OS Release 10.2.</p>
Description	<p>Define a list of IPv4 or IPv6 address prefixes for use in a routing policy statement or firewall filter statement.</p> <p>You can configure up to 85,325 prefixes in each prefix list. To configure more than 85,325 prefixes, configure multiple prefix lists and apply them to multiple firewall filter terms.</p>
Options	<p>name—Name that identifies the list of IPv4 or IPv6 address prefixes.</p> <p>ip-addresses—List of IPv4 or IPv6 address prefixes, one IP address per line in the configuration.</p> <p>The remaining statement is explained separately.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Configuring Prefix Lists for Use in Routing Policy Match Conditions Configuring Routing Policies and Policy Objects in the Dynamic Database dynamic-db "Firewall Filter Match Conditions Based on Address Fields on page 32" in the Junos OS Firewall Filter and Policer Configuration Guide "Example: Configuring a Filter to Limit TCP Access to a Port Based On a Prefix List on page 100" in the Junos OS Firewall Filter and Policer Configuration Guide

service-filter

Syntax	<pre>service-filter <i>filter-name</i> { term <i>term-name</i> { from { <i>match-conditions</i>; } then { <i>actions</i>; } } }</pre>
Hierarchy Level	[edit firewall family (inet inet6), [edit logical-systems <i>logical-system-name</i> firewall family (inet inet6)]
Release Information	Statement introduced before Junos OS Release 7.4. Logical systems support introduced in Junos OS Release 9.3.
Description	Configure service filters.
Options	<p><i>filter-name</i>—Name that identifies the service filter. The name can contain letters, numbers, and hyphens (-) and can be up to 255 characters long. To include spaces in the name, enclose it in quotation marks (" ").</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	firewall—To view this statement in the configuration. firewall-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">Guidelines for Configuring Service Filters on page 64Guidelines for Applying Service Filters on page 66

simple-filter

Syntax	<pre> simple-filter <i>filter-name</i> { term <i>term-name</i> { from { <i>match-conditions</i>; } then { <i>actions</i>; } } } </pre>
Hierarchy Level	[edit firewall family inet], [edit logical-systems <i>logical-system-name</i> firewall family inet]
Release Information	Statement introduced in Junos OS Release 7.6. Logical systems support introduced in Junos OS Release 9.3.
Description	Configure simple filters.
Options	<p><i>filter-name</i>—Name that identifies the simple filter. The name can contain letters, numbers, and hyphens (-) and can be up to 255 characters long. To include spaces in the name, enclose it in quotation marks (" ").</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	firewall—To view this statement in the configuration. firewall-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none"> Guidelines for Configuring Simple Filters on page 70 Guidelines for Applying Simple Filters on page 74

term

Syntax	<pre> term <i>term-name</i> { from { <i>match-conditions</i>; ip-version ipv4 { <i>match-conditions-mpls-ipv4-address</i>; protocol (tcp udp) { <i>match-conditions-mpls-ipv4-port</i>; } } } then { <i>actions</i>; } } </pre>
Hierarchy Level	<pre> [edit firewall family <i>family-name</i> filter <i>filter-name</i>], [edit firewall family <i>family-name</i> service-filter <i>filter-name</i>], [edit firewall family <i>family-name</i> simple-filter <i>filter-name</i>], [edit logical-systems <i>logical-system-name</i> firewall family <i>family-name</i> filter <i>filter-name</i>], [edit logical-systems <i>logical-system-name</i> firewall family <i>family-name</i> service-filter <i>filter-name</i>], [edit logical-systems <i>logical-system-name</i> firewall family <i>family-name</i> simple-filter <i>filter-name</i>] </pre>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>filter option introduced in Junos OS Release 7.6.</p> <p>Logical systems support introduced in Junos OS Release 9.3.</p> <p>ip-version ipv4 support introduced in Junos OS Release 10.1.</p>
Description	Define a firewall filter term.
Options	<p>actions—(Optional) Actions to perform on the packet if conditions match. You can specify one <i>terminating action</i> supported for the specified filter type. If you do not specify a terminating action, the packets that match the conditions in the from statement are accepted by default. As an option, you can specify one or more <i>nonterminating actions</i> supported for the specified filter type.</p> <p>filter-name—(Optional) For family <i>family-name</i> filter <i>filter-name</i> only, reference another standard stateless firewall filter from within this term.</p> <p>from—(Optional) Match packet fields to values. If not included, all packets are considered to match and the actions and action modifiers in the then statement are taken.</p> <p>match-conditions—One or more conditions to use to make a match on a packet.</p> <p>match-conditions-mpls-ipv4-address—(MPLS-tagged IPv4 traffic only) One or more IP address match conditions to match on the IPv4 packet header. Supports network-based service in a core network with IPv4 packets as an inner payload of an MPLS packet with labels stacked up to five deep.</p>

match-conditions-mpls-ipv4-port—(MPLS-tagged IPv4 traffic only) One or more UDP or TCP port match conditions to use to match a packet in an MPLS flow. Supports network-based service in a core network with IPv4 packets as an inner payload of an MPLS packet with labels stacked up to five deep.

term-name—Name that identifies the term. The name can contain letters, numbers, and hyphens (-) and can be up to 64 characters long. To include spaces in the name, enclose it in quotation marks (" ").

then—(Optional) Actions to take on matching packets. If not included and a packet matches all the conditions in the ***from*** statement, the packet is accepted.

Required Privilege	firewall—To view this statement in the configuration.
Level	firewall-control—To add this statement to the configuration.

Related Documentation	<ul style="list-style-type: none">• Guidelines for Configuring Standard Firewall Filters on page 17• Guidelines for Configuring Service Filters on page 64• Guidelines for Configuring Simple Filters on page 70• Guidelines for Configuring and Applying Firewall Filters in Logical Systems on page 78
------------------------------	---

PART 4

Index

- Index on page 315

Index

Symbols

! (negation)	
in firewall filters	
bit-field logical operator.....	28
&, bit-field logical operator.....	28
+	
bit-field logical operator.....	28
, (comma), bit-field logical operator.....	28
(pipe)	
in firewall filters	
bit-field logical operator.....	28

A

accounting	
standard stateless firewall filters	
applying firewall filter accounting	
profiles.....	300
configuring firewall filter accounting	
profiles.....	299
example.....	201
overview.....	55
accounting-profile statement.....	301
actions, flow control.....	21
actions, nonterminating	
for service filters.....	282
for simple filters.....	73
for standard stateless firewall filters.....	270
actions, terminating	
for service filters.....	281
for simple filters.....	73
for standard stateless firewall filters.....	269
address class, source or destination	
stateless firewall filter match conditions	
IPv4 traffic.....	236
IPv6 traffic.....	245
overview.....	39
address prefix, source or destination	
stateless firewall filter match conditions	
MPLS-tagged IPv4 traffic.....	252
VPLS traffic.....	254

address, source or destination	
stateless firewall filter match conditions	
IPv4 traffic.....	236
IPv6 traffic.....	245
Layer 2 bridging traffic.....	263
VPLS traffic.....	254
ampersand (&), bit-field logical operator.....	28
apply-path statement	
firewall filter match condition.....	100

B

bit-field	
logical operators.....	28

C

configuration examples	
service filters.....	211
simple filter.....	217

D

denial-of-service attacks, preventing.....	151
destination MAC address	
stateless firewall filter match conditions	
Layer 2 CCC traffic.....	261
MPLS-tagged IPv4 traffic.....	252
VPLS traffic.....	254
diagnosis	
displaying stateless firewall filter	
configurations.....	128, 155, 165
displaying stateless firewall filter	
statistics.....	157
verifying firewall filter handles fragments.....	165
verifying stateless firewall filter actions.....	128
verifying stateless firewall filter DoS	
protection.....	156
verifying stateless firewall filter flood	
protection.....	156
verifying stateless firewall filters with packet	
logs.....	129
DoS (denial-of-service) attacks, preventing.....	151

DSCP code point	
stateless firewall filter match condition	
IPv4 traffic.....	236
Layer 2 bridging traffic.....	263
VPLS traffic.....	254
E	
exclamation point (!), bit-field logical	
operator.....	28
F	
family statement	
firewall filter.....	302
files	
firewall log output file.....	123
filter statement	
firewall.....	304
filter-based forwarding	
standard stateless firewall filters	
applying filters to interfaces.....	297
configuring for IPv4 or IPv6 traffic.....	294
configuring for MPLS-tagged IPv4	
traffic.....	295
example.....	194
overview.....	53
firewall filters	
configuring on logical systems.....	223
log output file.....	123
verifying fragment handling.....	165
firewall filters in logical systems	
restrictions	
references from nonfirewall filter	
objects.....	84
references to nonfirewall filter	
objects.....	82
references to subordinate objects.....	81
firewall log output file.....	123
firewall statement.....	305
flooding, preventing.....	151
forwarding class	
stateless firewall filter match conditions	
IPv4 traffic.....	236
IPv6 traffic.....	245
Layer 2 bridging traffic.....	263
Layer 2 CCC traffic.....	261
protocol-independent traffic.....	235
VPLS traffic.....	254
fxp0.....	66
H	
handling packet fragments.....	161
I	
ICMP (Internet Control Message Protocol),	
policers.....	151
interface groups	
filtering packets received on	
applying filters.....	292
assigning logical interfaces to groups.....	291
configuring filters.....	291
example.....	184
overview.....	52
interface set	
filtering packets received on	
configuring filters.....	293
defining the interfaces in the set.....	293
overview.....	53
interface-set statement.....	305
interface-specific counters	
example	
example.....	180
interface-specific firewall filter instances	
filtering packets received on	
guidelines for applying.....	290
guidelines for configuring.....	289
overview.....	51
interface-specific names	
filter instance.....	51
interface-specific statement.....	306
Internet Control Message Protocol policers.....	151
IPv4 traffic	
match conditions	
standard stateless firewall filters.....	236
service filter actions, nonterminating.....	282
service filter actions, terminating.....	281
service filter match conditions.....	275
stateless firewall filter match conditions	
protocol-independent traffic.....	235
IPv6 traffic	
match conditions	
standard stateless firewall filters.....	245
service filter actions, nonterminating.....	282
service filter actions, terminating.....	281
service filter match conditions.....	275
stateless firewall filter match conditions	
protocol-independent traffic.....	235

L

Layer 2 bridging traffic	
match conditions	
standard stateless firewall filters.....	263
Layer 2 CCC traffic	
match conditions	
standard stateless firewall filter.....	261
log output	
firewall filters.....	123
logging	
standard stateless firewall filters	
example.....	206
system logging of firewall facility	
events.....	56
system logging of packet headers.....	58
system logging overview.....	55
logical systems	
configuring firewall filters.....	223
restrictions for firewall filters	
references from nonfirewall filter	
objects.....	84
references to nonfirewall filter	
objects.....	82
references to subordinate objects.....	81
stateless firewall filters	
applying.....	78
configuring.....	78
overview.....	77
unsupported firewall filter actions.....	285
unsupported firewall filter statements.....	283
loopback interface, applying stateless firewall filters	
to (configuration editor).....	151
loss priority	
stateless firewall filter match conditions	
IPv4 traffic.....	236
IPv6 traffic.....	245
Layer 2 bridging traffic.....	263
Layer 2 CCC traffic.....	261
VPLS traffic.....	254

M

management interface.....	66
match condition categories	
stateless firewall filters	
matching on address classes.....	39
matching on address prefixes.....	32
matching on bit-field values.....	28
matching on numeric values.....	27
matching on text strings.....	27

match conditions	
for service filters.....	275
match conditions for standard stateless firewall	
filters	
IPv4 traffic.....	236
IPv6 traffic.....	245
Layer 2 bridging traffic.....	263
Layer 2 CCC traffic.....	261
MPLS traffic.....	250
MPLS-tagged IPv4 traffic.....	252
protocol-independent traffic.....	235
VPLS traffic.....	254
MPLS traffic	
match conditions	
standard stateless firewall filters.....	250
MPLS-tagged IPv4 traffic	
match conditions	
standard stateless firewall filters.....	252
multiple standard firewall filters	
applied as a list	
example.....	171
guidelines for applying.....	47
overview.....	43, 44
in a nested configuration	
example.....	176
guidelines for configuring.....	49
overview.....	48
multiple stateless firewall filters	
applied as a list	
filter list name.....	46

N

next term action.....	21
noncontiguous address filter.....	32

O

output files	
firewall log output file.....	123

P

packet evaluation	
service filters.....	62
simple filters.....	69
standard stateless firewall filters.....	16
packets	
handling packet fragments (configuration	
editor).....	161
ping command (stateless firewall filter).....	156
explanation.....	156

pipe ()	
bit-field logical operator.....	28
plus sign (+), bit-field logical operator.....	28
policers	
for stateless firewall filters.....	151
policy framework.....	3
policy, routing	
prefix list.....	307
port number (TCP or UDP), source or destination	
stateless firewall filter match conditions	
IPv4 traffic.....	236
IPv6 traffic.....	245
Layer 2 bridging traffic.....	263
MPLS-tagged IPv4 traffic.....	252
VPLS traffic.....	254
prefix list.....	307
prefix list statement	
firewall filter match condition.....	100
prefix-list statement.....	307
usage guidelines.....	32
protocol-independent traffic	
match conditions	
standard stateless firewall filters.....	235

R

reverse-path forwarding (RPF)	
stateless firewall filters	
example.....	141
with an input firewall log or count.....	120
router data flow.....	3
Routing Engine	
handling packet fragments for (configuration	
editor).....	161
protecting against DoS attacks.....	151
protecting against untrusted services and	
protocols (configuration editor).....	125
Routing Engine traffic from trusted sources	
stateless firewall filters	
accepting OSPF packets from addresses	
in a prefix.....	139
blocking Telnet and SSH access.....	130
blocking TFTP access.....	136
example: accepting DHCP packets with	
specific addresses.....	141
routing solutions	
filtering unwanted services and protocols.....	125
handling packet fragments (configuration	
editor).....	161
protecting against DoS attacks.....	151

RPF	
firewall log and count.....	120

S

sample configurations	
firewall filter configurations.....	128, 155, 165
service filters	
actions	
nonterminating.....	282
terminating.....	281
configuration example.....	211
filtering packets received on a set of interface	
groups	
configuring filters.....	291
guidelines for applying.....	66
guidelines for configuring.....	64
interface-specific counters	
example.....	180
guidelines for applying.....	290
guidelines for configuring.....	289
overview.....	51
interface-specific policers	
guidelines for applying.....	290
guidelines for configuring.....	289
overview.....	51
match conditions.....	275
overview.....	6, 61
packet evaluation.....	62
service-filter statement	
firewall.....	308
show firewall command.....	128, 155, 165
show firewall filter protect-RE command.....	157
show firewall log command.....	129
show interfaces lo0 command.....	151
show log command.....	123
show route summary command.....	129, 165
explanation.....	128
simple filters	
configuration example.....	217
guidelines for applying.....	74
guidelines for configuring.....	70
overview.....	7, 69
packet evaluation.....	69
simple-filter statement	
firewall.....	309
ssh command.....	128

standard stateless firewall filters	
accounting	
applying firewall filter accounting	
profiles.....	300
configuring firewall filter accounting	
profiles.....	299
example.....	201
overview.....	55
actions	
nonterminating.....	270
terminating.....	269
applying.....	22
configuring.....	17
actions.....	21
filter names and options.....	19
filter terms.....	19
match conditions.....	19
protocol families.....	18
examples	
filter-based forwarding.....	194
logging for standard stateless firewall filter	
term.....	206
filter-based forwarding	
configuring for IPv4 or IPv6 traffic.....	294
configuring for MPLS-tagged IPv4	
traffic.....	295
example.....	194
overview.....	53
filtering packets received on a set of interface	
groups	
assigning logical interfaces to groups.....	291
configuring filters.....	291
overview.....	52
filtering packets received on a specific interface	
group	
applying filters.....	292
example.....	184
filtering packets received on a specific interface	
set	
configuring filters.....	293
overview.....	53
filtering packets received on an interface set	
defining the interfaces in the set.....	293
interface-specific counters	
example.....	180
interface-specific policers	
guidelines for applying.....	290
guidelines for configuring.....	289
overview.....	51
logging	
example.....	206
system logging of firewall facility	
events.....	56
system logging of packet headers.....	58
system logging overview.....	55
multiple filters applied as a list	
example.....	171
guidelines for applying.....	47
overview.....	43, 44
multiple filters in a nested configuration	
example.....	176
guidelines for configuring.....	49
overview.....	48
overview.....	15
packet evaluation.....	16
standards	
supported for filtering.....	229
stateless firewall filter	
supported standards.....	229
stateless firewall filters	
accepting Routing Engine traffic from trusted	
sources	
example: blocking TCP access.....	144
example: blocking Telnet and SSH	
access.....	100, 158
actions.....	10
firewall filters in logical systems.....	78
service filters.....	64
standard stateless firewall filters.....	21
unsupported in logical systems.....	285
actions, nonterminating	
service filters.....	282
simple filters.....	73
standard stateless firewall filters.....	270
actions, terminating	
service filters.....	281
simple filters.....	73
standard stateless firewall filters.....	269
application points	
overview.....	11
service filters.....	66
simple filters.....	74
applying to an interface (configuration	
editor).....	151
basic use	
filtering data packets.....	24
filtering local packets.....	24
handling packet fragments.....	41

displaying configurations.....	128, 155, 165
displaying statistics.....	157
examples	
accepting DHCP packets with specific addresses.....	141
accepting OSPF packets from addresses in a prefix.....	139
accepting packets with specific IPv6 TCP flags.....	158
accounting for standard stateless firewall filters.....	201
applying lists of standard firewall filters to a single interface.....	171
blocking TCP access.....	144
blocking Telnet and SSH access.....	100, 130
blocking TFTP access.....	136
counting accepted and rejected packets.....	105
counting and discarding IP options packets.....	108
counting and sampling accepted packets.....	120
counting IP option packets.....	111
matching on destination port and protocol.....	94
matching on IPv6 flags.....	93
matching on unrelated fields.....	97
nesting references to multiple firewall filters.....	176
setting rate limits based on destination class.....	167
setting rate limits for traffic received on an interface set.....	188
setting the DSCP bit to zero.....	117
filter names	
service filters.....	64
simple filters.....	70
filter names and options	
standard stateless firewall filters.....	19
filter terms.....	9
service filters.....	64
simple filters.....	70
standard stateless firewall filters.....	19
filter-based forwarding	
applying filters to interfaces.....	297, 298
filtering router transit traffic	
overview.....	24
filtering Routing Engine traffic	
overview.....	24
firewall filter statements	
unsupported in logical systems.....	283
handling packet fragments	
overview.....	41
handling packet fragments (configuration editor).....	161
hardware requirements for applying	
service filters.....	66
simple filters.....	74
in logical systems	
applying.....	78
configuring.....	78
overview.....	77
interface-specific names	
filter list name.....	46
logical systems	
unsupported firewall filter actions.....	285
unsupported firewall filter statements.....	283
match condition categories	
matching on address classes.....	39
matching on address prefixes.....	32
matching on bit-field values.....	28
matching on numeric values.....	27
matching on text strings.....	27
match conditions.....	9
firewall filters in logical systems.....	78
service filters.....	64, 275
simple filters.....	70
standard stateless firewall filters.....	19
multiple filters applied as a list	
filter list name.....	46
overview.....	5
policers for.....	151
protecting the Routing Engine against TCP floods.....	151
protecting the Routing Engine against untrusted protocols (configuration editor).....	125
protecting the Routing Engine against untrusted services (configuration editor).....	125
protocol families.....	7
firewall filters in logical systems.....	78
service filters.....	64
simple filters.....	70
standard stateless firewall filters.....	18
reverse-path forwarding (RPF)	
example.....	141
with an input firewall log or count.....	120
sample terms, to filter fragments.....	161

- sample terms, to filter services and
 - protocols.....125
- service filters.....61
 - statement hierarchy for applying.....66
 - statement hierarchy for configuring.....64
- simple filters.....69
- standard stateless firewall filters.....15
- statement hierarchy
 - applying simple filters.....74
 - configuring simple filters.....70
- type
 - overview.....7
- types.....6
- verifying actions.....128
- verifying configuration.....128, 155, 165
- verifying flood protection.....156
- verifying packet logging.....129
- statement hierarchy
 - service filters
 - applying.....66
 - configuring.....64
 - simple filters
 - applying.....74
 - configuring70
 - stateless firewall filters
 - applying filters to interfaces.....298
- statistics
 - stateless firewall filters.....157
- T**
- TCP policers.....151
- telnet command.....156
- term statement
 - firewall.....310
- traffic
 - sampling
 - show log command.....123
- V**
- verification
 - firewall filter handles fragments.....165
 - OSPF policy.....226
 - stateless firewall filter actions.....128
 - stateless firewall filter flood protection.....156
 - stateless firewall filter operation.....129
 - stateless firewall filters.....128, 155, 165
 - stateless firewall statistics.....157
- VPLS traffic
 - match conditions
 - standard stateless firewall filters.....254

