

Application Properties



Published: 2013-08-29

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.

Application Properties

Copyright © 2013, Juniper Networks, Inc.
All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About the Documentation	ix
Documentation and Release Notes	ix
Supported Platforms	ix
Using the Examples in This Manual	ix
Merging a Full Example	x
Merging a Snippet	x
Documentation Conventions	xi
Documentation Feedback	xiii
Requesting Technical Support	xiii
Self-Help Online Tools and Resources	xiii
Opening a Case with JTAC	xiv

Part 1

Chapter 1

Overview

Application-Level Gateway	3
ALG Descriptions	3
Supported ALGs	3
ALG Support Details	4
Basic TCP ALG	5
Basic UDP ALG	5
BOOTP	6
DCE RPC Services	6
DNS	6
FTP	6
H323	7
ICMP	7
IIOP	8
IP	8
NetBIOS	8
NetShow	8
ONC RPC Services	9
PPTP	9
RealAudio	9
Sun RPC and RPC Portmap Services	10
RTSP	11
SIP	11
SNMP	12
SQLNet	12
TFTP	12
Traceroute	13
UNIX Remote-Shell Services	13

	Winframe	13
	Juniper Networks Defaults	14
	Junos Default Groups	23
	Examples: Referencing the Preset Statement from the Junos Default Group	29
	ALGs Available by Default for Junos OS Address Aware NAT	30
Part 2	Configuration	
Chapter 2	Configuration Tasks	37
	Configuring Application Sets	37
	Configuring Application Protocol Properties	37
	Configuring an Application Protocol	38
	Configuring the Network Protocol	40
	Configuring the ICMP Code and Type	41
	Configuring Source and Destination Ports	43
	Configuring the Inactivity Timeout Period	46
	Configuring SIP	46
	SIP ALG Interaction with Network Address Translation	47
	Junos OS SIP ALG Limitations	52
	Configuring an SNMP Command for Packet Matching	53
	Configuring an RPC Program Number	53
	Configuring the TTL Threshold	54
	Configuring a Universal Unique Identifier	54
Chapter 3	Example	55
	Examples: Configuring Application Protocols	55
Chapter 4	Verification	57
	Verifying the Output of ALG Sessions	57
	FTP Example	57
	Sample Output	57
	FTP System Log Messages	58
	Analysis	59
	Troubleshooting Questions	59
	RTSP ALG Example	60
	Sample Output	60
	Analysis	60
	Troubleshooting Questions	60
	System Log Messages	62
	System Log Configuration	62
	System Log Output	63
Chapter 5	Configuration Statements	65
	application	65
	application-protocol	66
	application-set	67
	applications	68
	destination-port	68
	icmp-code	69

icmp-type	69
inactivity-timeout	70
learn-sip-register	70
protocol	71
rpc-program-number	72
sip-call-hold-timeout	72
snmp-command	73
source-port	73
ttl-threshold	74
uuid	74

Part 3

Index

Index	77
-------------	----

List of Tables

	About the Documentation	ix
	Table 1: Notice Icons	xi
	Table 2: Text and Syntax Conventions	xi
Part 1	Overview	
Chapter 1	Application-Level Gateway	3
	Table 3: ALGs Supported by the Junos OS	3
	Table 4: RealAudio Product Port Usage	9
	Table 5: Supported RPC Services	10
	Table 6: ALGs Available by Default	31
Part 2	Configuration	
Chapter 2	Configuration Tasks	37
	Table 7: Application Protocols Supported by Services Interfaces	38
	Table 8: Network Protocols Supported by Services Interfaces	40
	Table 9: ICMP Codes and Types Supported by Services Interfaces	42
	Table 10: Port Names Supported by Services Interfaces	43
	Table 11: Requesting Messages with NAT Table	51

About the Documentation

- [Documentation and Release Notes on page ix](#)
- [Supported Platforms on page ix](#)
- [Using the Examples in This Manual on page ix](#)
- [Documentation Conventions on page xi](#)
- [Documentation Feedback on page xiii](#)
- [Requesting Technical Support on page xiii](#)

Documentation and Release Notes

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

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at <http://www.juniper.net/books>.

Supported Platforms

For the features described in this document, the following platforms are supported:

- [M Series](#)
- [T Series](#)
- [MX Series](#)
- [J Series](#)

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

```
system {
  scripts {
    commit {
      file ex-script.xml;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

```
[edit]
user@host# load merge /var/tmp/ex-script.conf
load complete
```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {
  file ex-script-snippet.xml; }
```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see the *CLI User Guide*.

Documentation Conventions

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

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> show chassis alarms No alarms currently active

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

Convention	Description	Examples
<i>Italic text like this</i>	<ul style="list-style-type: none">Introduces or emphasizes important new terms.Identifies guide names.Identifies RFC and Internet draft titles.	<ul style="list-style-type: none">A policy <i>term</i> is a named structure that defines match conditions and actions.<i>Junos OS CLI User Guide</i>RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none">To configure a stub area, include the stub statement at the [edit protocols ospf area <i>area-id</i>] hierarchy level.The console port is labeled CONSOLE.
< > (angle brackets)	Enclose optional keywords or variables.	stub <default-metric <i>metric</i>>;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Enclose a variable for which you can substitute one or more values.	community name members [<i>community-ids</i>]
Indentation and braces ({ })	Identify a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none">In the Logical Interfaces box, select All Interfaces.To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

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

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

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://www.juniper.net/alerts/>

- Join and participate in the Juniper Networks Community Forum:
<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

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

Opening a Case with JTAC

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

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

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

PART 1

Overview

- [Application-Level Gateway on page 3](#)

CHAPTER 1

Application-Level Gateway

- [ALG Descriptions on page 3](#)
- [Junos Default Groups on page 23](#)
- [ALGs Available by Default for Junos OS Address Aware NAT on page 30](#)

ALG Descriptions

This topic describes the Application Layer Gateways (ALGs) supported by the Junos OS. ALG support includes managing pinholes and parent-child relationships for the supported all ALGs. This topic includes the following section:

- [Supported ALGs on page 3](#)
- [ALG Support Details on page 4](#)
- [Juniper Networks Defaults on page 14](#)

Supported ALGs

[Table 3 on page 3](#) lists ALGs supported by the Junos OS.

Table 3: ALGs Supported by the Junos OS

ALGs Supported	v4 - v4	v4 - v6	v6 - v6	DS-Lite
Basic TCP ALG	Yes	Yes	Yes	Yes
Basic UPD ALG	Yes	Yes	Yes	Yes
BOOTP	Yes	No	No	No
DCE RPC Services	Yes	No	No	No
DNS	Yes	Yes	No	No
FTP	Yes	No	No	Yes
H323	Yes	No	No	No
ICMP	Yes	Yes	Yes	Yes

Table 3: ALGs Supported by the Junos OS (*continued*)

ALGs Supported	v4 - v4	v4 - v6	v6 - v6	DS-Lite
IIOIP	Yes	No	No	No
IP	Yes	No	No	No
NETBIOS	Yes	No	No	No
NETSHOW	Yes	No	No	No
PPTP	Yes	No	No	Yes
REALAUDIO	Yes	No	No	No
Sun RPC and RPC Port Map Services	Yes	No	No	No
RTSP	Yes	No	No	Yes
SIP	Yes	No	No	No
SNMP	Yes	No	No	No
SQLNET	Yes	No	No	No
TFTP	Yes	No	No	Yes
Traceroute	Yes	Yes	No	Yes
Unix Remote Shell Service	Yes	No	No	No
WINFrame	Yes	No	No	No

ALG Support Details

This section includes details about the ALGs. It includes the following:

- [Basic TCP ALG on page 5](#)
- [Basic UDP ALG on page 5](#)
- [BOOTP on page 6](#)
- [DCE RPC Services on page 6](#)
- [DNS on page 6](#)
- [FTP on page 6](#)
- [H323 on page 7](#)
- [ICMP on page 7](#)
- [IIOIP on page 8](#)

- [IP on page 8](#)
- [NetBIOS on page 8](#)
- [NetShow on page 8](#)
- [ONC RPC Services on page 9](#)
- [PPTP on page 9](#)
- [RealAudio on page 9](#)
- [Sun RPC and RPC Portmap Services on page 10](#)
- [RTSP on page 11](#)
- [SIP on page 11](#)
- [SNMP on page 12](#)
- [SQLNet on page 12](#)
- [TFTP on page 12](#)
- [Traceroute on page 13](#)
- [UNIX Remote-Shell Services on page 13](#)
- [Winframe on page 13](#)

Basic TCP ALG

This ALG performs basic sanity checking on TCP packets. If it finds errors, it generates the following anomaly events and system log messages:

- TCP source or destination port zero
- TCP header length check failed
- TCP sequence number zero and no flags are set
- TCP sequence number zero and FIN/PSH/RST flags are set
- TCP FIN/RST or SYN(URG|FIN|RST) flags are set

The TCP ALG performs the following steps:

1. When the router receives a SYN packet, the ALG creates TCP forward and reverse flows and groups them in a *conversation*. It tracks the TCP three-way handshake.
2. The SYN-defense mechanism tracks the TCP connection establishment state. It expects the TCP session to be established within a small time interval (currently 4 seconds). If the TCP three-way handshake is not established in that period, the session is terminated.
3. A keepalive mechanism detects TCP sessions with nonresponsive endpoints.
4. ICMP errors are allowed only if there is a flow that matches the selector information specified in the ICMP data.

Basic UDP ALG

This ALG performs basic sanity checking on UDP headers. If it finds errors, it generates the following anomaly events and system log messages:

- UDP source or destination port 0
- UDP header length check failed

The UDP ALG performs the following steps:

1. When it receives the first packet, the ALG creates bidirectional flows to accept forward and reverse UDP session traffic.
2. If the session is idle for more than the maximum allowed idle time (the default is 30 seconds), the flows are deleted.
3. ICMP errors are allowed only if there is a flow that matches the selector information specified in the ICMP data.

BOOTP

The Bootstrap Protocol (BOOTP) client retrieves its networking information from a server across the network. It sends out a general broadcast message to request the information, which is returned by the BOOTP server. For the protocol specification, see <ftp://ftp.isi.edu/in-notes/rfc951.txt>.

Stateful firewall support requires that you configure the BOOTP ALG on UDP server port 67 and client port 68. If the client sends a broadcast message, you should configure the broadcast address in the **from** statement of the service rule. Network Address Translation (NAT) is not performed on the BOOTP traffic, even if the NAT rule matches the traffic. If the BOOTP relay feature is activated on the router, the remote BOOTP server is assumed to assign addresses for clients masked by NAT translation.

DCE RPC Services

Distributed Computing Environment (DCE) Remote Procedure Call (RPC) services are mainly used by Microsoft applications. The ALG uses well-known TCP port 135 for port mapping services, and uses the universal unique identifier (UUID) instead of the program number to identify protocols. The main application-based DCE RPC is the Microsoft Exchange Protocol.

Support for stateful firewall and NAT services requires that you configure the DCE RPC portmap ALG on TCP port 135. The DCE RPC ALG uses the TCP protocol with application-specific UUIDs.

DNS

The Domain Name Service (DNS) ALG handles data associated with locating and translating domain names into IP addresses. The ALG typically runs on port 53. The ALG monitors DNS query and reply packets and supports only UDP traffic. The ALG does not support payload translations. The DNS ALG will only close the session when a reply is received or an idle timeout is reached.

FTP

FTP is the File Transfer Protocol, specified in RFC 959. In addition to the main control connection, data connections are also made for any data transfer between the client

and the server; and the host, port, and direction are negotiated through the control channel.

For non-passive-mode FTP, the Junos OS stateful firewall service scans the client-to-server application data for the PORT command, which provides the IP address and port number to which the server connects. For passive-mode FTP, the Junos OS stateful firewall service scans the client-to-server application data for the PASV command and then scans the server-to-client responses for the 227 response, which contains the IP address and port number to which the client connects.

There is an additional complication: FTP represents these addresses and port numbers in ASCII. As a result, when addresses and ports are rewritten, the TCP sequence number might be changed, and thereafter the NAT service needs to maintain this delta in SEQ and ACK numbers by performing sequence NAT on all subsequent packets.

Support for stateful firewall and NAT services requires that you configure the FTP ALG on TCP port 21 to enable the FTP control protocol. The ALG performs the following tasks:

- Automatically allocates data ports and firewall permissions for dynamic data connection
- Creates flows for the dynamically negotiated data connection
- Monitors the control connection in both active and passive modes
- Rewrites the control packets with the appropriate NAT address and port information

H323

H323 is a suite of ITU protocols for audio and video conferencing and collaboration applications. H323 consists of H.225 call signaling protocols and H.245 control protocol for media communication. During H.225 negotiation, the endpoints create a call by exchanging call signaling messages on the control channel and negotiate a new control channel for H.245. A new control connection is created for H.245 messages. Messages are exchanged on the H.245 control channel to open media channels.

Stateful firewall monitors the H.225 control channel to open the H.245 control channel. After the H.245 channel is created, stateful firewall also monitors this channel for media channel information and allows the media traffic through the firewall.

H323 ALG supports static destination, static and dynamic source NAT by rewriting the appropriate addresses and ports in the H.225 and H.245 messages.

ICMP

The Internet Control Message Protocol (ICMP) is defined in RFC 792. The Junos OS stateful firewall service allows ICMP messages to be filtered by specific type or specific type code value. ICMP error packets that lack a specifically configured type and code are matched against any existing flow in the opposite direction to check for the legitimacy of the error packet. ICMP error packets that pass the filter matching are subject to NAT translation.

The ICMP ALG always tracks ping traffic statefully using the ICMP sequence number. Each echo reply is forwarded only if there is an echo request with the corresponding sequence number. For any ping flow, only 20 echo requests can be forwarded without receiving an echo reply. When you configure dynamic NAT, the PING packet identifier is translated to allow additional hosts in the NAT pool to use the same identifier.

Support for stateful firewall and NAT services requires that you configure the ICMP ALG if the protocol is needed. You can configure the ICMP type and code for additional filtering.

IIOIP

The Oracle Application Server NameServer Internet Inter-ORB Protocol (IIOIP). This ALG is used in Common Object Request Broker Architecture (CORBA) based on distributed computing. Even though CORBA and IIOIP are Object Management Group (OMG) standards, there is no fixed port assigned for IIOIP. Each vendor implementing CORBA chooses a port. Java Virtual machine uses port 1975 by default, while ORBIX uses port 3075 as a default.

Stateful firewall and NAT require ALG IIOIP be configured for TCP port 1975 for Java VM IIOIP, and 3075 for CORBA applications ORBIX, a CORBA framework from Iona Technologies.

IP

The IP ALG is used to create uni-directional flows only. In case of TCP traffic, it does not check the 3-way handshake process. This ALG is useful in case of stateful firewall only service sets, where it allows traffic to flow uni-directionally only. When configuring in conjunction with **match-direction input-output** it allows the return traffic to flow through the stateful firewall as well. Typical scenarios are static NAT, destination NAT or scenarios where traffic is expected to traverse the stateful firewall in the presence of asymmetric routing. The Junos IP ALG is not intended for use with NAPT, which will cause matching traffic to be discarded through the creation of a drop flow.

NetBIOS

A NetBIOS ALG translates NetBIOS IP addresses and port numbers when NAT is used.

NetBIOS supports the TCP and UDP transport protocols. Support for stateful firewall and NAT services requires that you configure the NetBIOS ALG on UDP port 138 and TCP port 139.

NetShow

The Microsoft protocol ms-streaming is used by NetShow, the Microsoft media server. This protocol supports several transport protocols: TCP, UDP, and HTTP. The client starts a TCP connection on port 1755 and sends the PORT command to the server. The server then starts UDP on that port to the client. Support for stateful firewall and NAT services requires that you configure the NetShow ALG on UDP port 1755.

ONC RPC Services

Open Networks Computing (ONC) RPC services function similarly to DCE RCP services. However, the ONC RPC ALG uses TCP/UDP port 111 for port mapping services, and uses the program number to identify protocols rather than the UUID.

Support for stateful firewall and NAT services requires that you configure the ONC RPC portmap ALG on TCP port 111. The ONC RPC ALG uses the TCP protocol with application-specific program numbers.

PPTP

The Point-to-Point Tunneling Protocol (PPTP) ALG is a TCP-based ALG. PPTP allows the Point-to-Point Protocol (PPP) to be tunneled through an IP network. PPTP defines a client-server architecture, a PPTP Network Server, and a PPTP Access Concentrator. The PPTP ALG requires a control connection and a data tunnel. The control connection uses TCP to establish and disconnect PPP sessions, and runs on port 1723. The data tunnel carries PPP traffic in generic routing encapsulated (GRE) packets that are carried over IP.

RealAudio

Real Networks PNA protocol RealVideo is not a separate service. It is part of the RealPlayer and most likely uses another channel for video. The RealPlayer versions G2, 7, and 8 use PNA and RTSP. For this version to work, the ALG must allow both PNA(7070) and RTSP(554). For the media, the server selects from a range of UDP ports(6970 through 7170), or TCP port 7071, or HTTP. The client can be configured to use a particular port. The RealPlayer versions 4.0 and 5.0 use control channel 7070 media UDP ports 6970 through 7170, or TCP port 7071, or HTTP. RealAudio player version 3.0 uses control channel 7070 media, UDP ports 6770-7170, or TCP port 7071.

Real products use the ports and ranges of ports shown in [Table 4 on page 9](#).

Table 4: RealAudio Product Port Usage

Real Product	Port Usage
4.0 and 5.0 Servers/Players	Control channel (bidirectional) on TCP port 7070. Data channel from server to player on TCP port 7070 or UDP port 6970-7170.
4.0 and 5.0 Servers/Encoders	Control channel (bidirectional) on TCP port 7070. Data channel from encoder or server on TCP port 7070.
G2 Servers/Players	Control channel (bidirectional) on TCP port 80, 554, 7070, or 8080. Data channel from server to player on TCP port 80, 554, 7070, 8080 or UDP port 6970-32,000.
G2 Server/3.1, and 5.x Encoders	Control channel (bidirectional) on TCP port 7070. Data channel from encoder to server on TCP port 7070.
G2 Server/G2 Producer	Control channel (bidirectional) on TCP port 4040. Data channel from encoder to server on TCP port 4040 and UDP port 6970-32,000.

Table 4: RealAudio Product Port Usage (*continued*)

Real Product	Port Usage
2 Server/G2 Producer (TCP ONLY)	Control channel (bidirectional) on TCP port 4040 Data channel from encoder to server on TCP port 4040. Note: TCP-ONLY option available in version 6.1 or above.



NOTE: RealAudio was the original protocol by RealPlayers. Newer versions of RealPlayer use RTSP. Stateful firewall and NAT require ALG RealAudio to be programmed on TCP port 7070.

Sun RPC and RPC Portmap Services

The Remote Procedure Call (RPC) ALG uses well-known ports TCP 111 and UDP 111 for port mapping, which dynamically assigns and opens ports for RPC services. The RPC Portmap ALG keeps track of port requests and dynamically opens the firewall for these requested ports. The RPC ALG can further restrict the RPC protocol by specifying allowed program numbers.

The ALG includes the RPC services listed in [Table 5 on page 10](#).

Table 5: Supported RPC Services

Name	Description	Comments
rpc-mountd	Network File Server (NFS) mount daemon; for details, see the UNIX man page for rpc.mountd(8) .	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050).
rpc-nfsprog	Used as part of NFS. For details, see RFC 1094. See also RFC1813 for NFS v3.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050).
rpc-nisplus	Network Information Service Plus (NIS+), designed to replace NIS; it is a default naming service for Sun Solaris and is not related to the old NIS. No protocol information is available.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050).
rpc-nlockmgr	Network lock manager.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, rpc-nlockmgr service can be allowed or blocked based on RPC program 100021.
rpc-pcnfsd	Kernel statistics server. For details, see the UNIX man pages for rstatd and rpc.rstatd .	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, rpc-rstat service can be allowed or blocked based on RPC program 150001.
rpc-rwall	Used to write a message to users; for details, see the UNIX man page for rpc.rwalld .	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, rpc-rwall service can be allowed or blocked based on RPC program 150008.

Table 5: Supported RPC Services (*continued*)

Name	Description	Comments
rpc-yplib	NIS binding process. For details, see the UNIX man page for yplib .	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, rpc-yplib service can be allowed or blocked based on RPC program 100007.
rpc-yppasswd	NIS password server. For details, see the UNIX man page for yppasswd .	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, rpc-yppasswd service can be allowed or blocked based on RPC program 100009.
rpc-ypserv	NIS server. For details, see the UNIX man page for ypserv .	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, rpc-ypserv service can be allowed or blocked based on RPC program 100004.
rpc-ypupdated	Network updating tool.	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, rpc-ypupdated service can be allowed or blocked based on RPC program 100028.
rpc-ypxfrd	NIS map transfer server. For details, see the UNIX man page for rpc.ypxfrd .	The base support is RPC v2 and the port mapper service on port 111 (see RFC 1050). Once the RPC program table is built, rpc-ypxfrd service can be allowed or blocked based on RPC program 100069.

Support for stateful firewall and NAT services that use port mapping requires that you configure the RPC portmap ALG on TCP/UDP destination port 111 and the RPC ALG for both TCP and UDP. You can specify one or more **rpc-program-number** values to further restrict allowed RPC protocols.

RTSP

The Real-Time Streaming Protocol (RTSP) controls the delivery of data with real-time properties such as audio and video. The streams controlled by RTSP can use RTP, but it is not required. Media can be transmitted on the same RTSP control stream. This is an HTTP-like text-based protocol, but client and server maintain session information. A session is established using the SETUP message and terminated using the TEARDOWN message. The transport (the media protocol, address, and port numbers) is negotiated in the setup and the setup-response.

Support for stateful firewall and NAT services requires that you configure the RTSP ALG for TCP port 554.

The ALG monitors the control connection, opens flows dynamically for media (RTP/RTSP) streams, and performs NAT address and port rewrites.

SIP

The Session Initiation Protocol (SIP) is an application layer protocol that can establish, maintain, and terminate media sessions. It is a widely used voice over IP (VoIP) signaling

protocol. The SIP ALG monitors SIP traffic and dynamically creates and manages pinholes on the signaling and media paths. The ALG only allows packets with the correct permissions. The SIP ALG also performs the following functions:

- Manages parent-child session relationships.
- Enforces security policies.
- Manages pinholes for VoIP traffic.

The SIP ALG supports the following features:

- Stateful firewall
- Static source NAT
- Dynamic address only source NAT
- Network Address Port Translation (NAPT)

SNMP

SNMP is a communication protocol for managing TCP/IP networks, including both individual network devices and aggregated devices. The protocol is defined by RFC 1157. SNMP runs on top of UDP.

The Junos OS stateful firewall service implements the SNMP ALG to inspect the SNMP type. SNMP does not enforce stateful flow. Each SNMP type needs to be specifically enabled. Full SNMP support of stateful firewall services requires that you configure the SNMP ALG on UDP port 161. This enables the SNMP **get** and **get-next** commands, as well as their response traffic in the reverse direction: UDP port 161 enables the SNMP **get-response** command. If SNMP traps are permitted, you can configure them on UDP port 162, enabling the SNMP **trap** command.

SQLNet

The SQLNet protocol is used by Oracle SQL servers to execute SQL commands from clients, including load balancing and application-specific services.

Support of stateful firewall and NAT services requires that you configure the SQLNet ALG for TCP port 1521.

The ALG monitors the control packets, opens flows dynamically for data traffic, and performs NAT address and port rewrites.

TFTP

The Trivial File Transfer Protocol (TFTP) is specified in RFC 1350. The initial TFTP requests are sent to UDP destination port 69. Additional flows can be created to **get** or **put** individual files. Support of stateful firewall and NAT services requires that you configure the TFTP ALG for UDP destination port 69.

Traceroute

Traceroute is a tool for displaying the route that packets take to a network host. It uses the IP time-to-live (TTL) field to trigger ICMP time-exceeded messages from routers or gateways. It sends UDP datagrams to destination ports that are believed to be not in use; destination ports are numbered using the formula: $+ n\text{hops} - 1$. The default base port is 33434. To support traceroute through the firewall, two types of traffic must be passed through:

1. UDP probe packets (UDP destination port > 33000 , IP TTL < 30)
2. ICMP response packets (ICMP type time-exceeded)

When NAT is applied, the IP address and port within the ICMP error packet also must be changed.

Support of stateful firewall and NAT services requires you to configure the Traceroute ALG for UDP destination port 33434 to 33450. In addition, you can configure the TTL threshold to prevent UDP flood attacks with large TTL values.

UNIX Remote-Shell Services

Three protocols form the basis for UNIX remote-shell services:

- Exec—Remote command execution; enables a user on the client system to execute a command on the remote system. The first command from client (**rcmd**) to server (**rshd**) uses well-known TCP port 512. A second TCP connection can be opened at the request of **rcmd**. The client port number for the second connection is sent to the server as an ASCII string.
- Login—Better known as **rlogin**; uses well-known TCP port 513. For details, see RFC 1282. No special firewall processing is required.
- Shell—Remote command execution; enables a user on the client system to execute a command on the remote system. The first command from client (**rcmd**) to server (**rshd**) uses well-known TCP port 514. A second TCP connection can be opened at the request of **rcmd**. The client port number for the second connection is sent to the server as an ASCII string.

Support of stateful firewall services requires that you configure the Exec ALG on TCP port 512, the Login ALG on TCP port 513, and the Shell ALG on TCP port 514. NAT remote-shell services require that any dynamic source port assigned be within the port range 512 to 1023. If you configure a NAT pool, this port range is reserved exclusively for remote shell applications.

Winframe

WinFrame application server software provides access to virtually any Windows application, across any type of network connection to any type of client.

This protocol is mainly used by Citrix Windows applications.

Stateful firewall and NAT require the ALG Winframe to be configured on TCP destination port 1494 and UDP port 1604.

Juniper Networks Defaults

```
applications {
  #
  # File Transfer Protocol
  #
  application junos-ftp {
    application-protocol ftp;
    protocol tcp;
    destination-port 21;
  }
  #
  # Trivial File Transfer Protocol
  #
  application junos-tftp {
    application-protocol tftp;
    protocol udp;
    destination-port 69;
  }
  #
  # RPC portmapper on TCP
  #
  application junos-rpc-portmap-tcp {
    application-protocol rpc-portmap;
    protocol tcp;
    destination-port 111;
  }
  #
  # RPC portmapper on UDP
  #
  application junos-rpc-portmap-udp {
    application-protocol rpc-portmap;
    protocol udp;
    destination-port 111;
  }
  #
  # SNMP get
  #
  application junos-snmp-get {
    application-protocol snmp;
    protocol udp;
    destination-port 161;
    snmp-command get;
  }
  #
  # SNMP get next
  #
  application junos-snmp-get-next {
    application-protocol snmp;
    protocol udp;
    destination-port 161;
    snmp-command get-next;
  }
}
```

```
#
# SNMP response
#
application junos-snmp-response {
    application-protocol snmp;
    protocol udp;
    source-port 161;
    snmp-command get-response;
}
#
# SNMP trap
#
application junos-snmp-trap {
    application-protocol snmp;
    protocol udp;
    destination-port 162;
    snmp-command trap;
}
#
# remote exec
#
application junos-rexec {
    application-protocol exec;
    protocol tcp;
    destination-port 512;
}
#
# remote login
#
application junos-rlogin {
    application-protocol shell;
    protocol tcp;
    destination-port 513;
}
#
# remote shell
#
application junos-rsh {
    application-protocol shell;
    protocol tcp;
    destination-port 514;
}
#
# Real Time Streaming Protocol
#
application junos-rtsp {
    application-protocol rtsp;
    protocol tcp;
    destination-port 554;
}
#
# Citrix windows application server protocol
# windows applications remotely on windows/non-windows clients
#
# citrix needs udp 1604 to be open
#
```

```
application junos-citrix-winframe {
    application-protocol winframe;
    protocol tcp;
    destination-port 1494;
}
application junos-citrix-winframe-udp {
    protocol udp;
    destination-port 1604;
}
#
# Oracle SQL servers use this protocol to execute sql commands
# from clients, load balance, use application-specific servers, etc
#
application junos-sqlnet {
    application-protocol sqlnet;
    protocol tcp;
    destination-port 1521;
}
#
# H.323 Protocol for audio/video conferencing
#
application junos-h323 {
    application-protocol h323;
    protocol tcp;
    destination-port 1720;
}
#
# Internet Inter-ORB Protocol - used for CORBA applications
# The ORB protocol in Java virtual machines uses port 1975 as default
#
application junos-iiop-java {
    application-protocol iiop;
    protocol tcp;
    destination-port 1975;
}
#
# Internet Inter-ORB Protocol - used for CORBA applications
# ORBIX is a CORBA framework from Iona Technologies that uses port
# 3075 as default
#
application junos-iiop-orbix {
    application-protocol iiop;
    protocol tcp;
    destination-port 3075;
}
#
# Real players use this protocol for real time streaming
# This was the original protocol for real players.
# RTSP is more widely used by real players
# but they still support realaudio.
#
application junos-realaudio {
    application-protocol realaudio;
    protocol tcp;
    destination-port 7070;
}
```

```
#
# traceroute application.
#
application junos-traceroute {
    application-protocol traceroute;
    protocol udp;
    destination-port 33435-33450;
    ttl-threshold 30;
}
#
# The full range of known RPC programs using UDP
# The program numbers can be more specific to certain applications.
#
application junos-rpc-services-udp {
    application-protocol rpc;
    protocol udp;
    rpc-program-number 100000-400000;
}
#
# The full range of known RPC programs using TCP
# The program numbers can be more specific to certain applications.
#
application junos-rpc-services-tcp {
    application-protocol rpc;
    protocol tcp;
    rpc-program-number 100000-400000;
}
#
# All ICMP traffic
# This can be made to be more restrictive by specifying ICMP type
# and code.
#
application junos-icmp-all {
    application-protocol icmp;
}
#
# Protocol used by Windows media server and windows media player
#
application junos-netshow {
    application-protocol netshow;
    protocol tcp;
    destination-port 1755;
}
#
# NetBIOS - networking protocol used on
# Windows networks name service port, both UDP and TCP
#
application junos-netbios-name-udp {
    application-protocol netbios;
    protocol udp;
    destination-port 137;
}
application junos-netbios-name-tcp {
    protocol tcp;
    destination-port 137;
}
```

```
#
# NetBIOS - networking protocol used on
# Windows networks datagram service port
#
application junos-netbios-datagram {
    application-protocol netbios;
    protocol udp;
    destination-port 138;
}
#
# NetBIOS - networking protocol used on
# Windows networks session service port
#
application junos-netbios-session {
    protocol tcp;
    destination-port 139;
}
#
# DCE-RPC portmapper on TCP
#
application junos-dce-rpc-portmap {
    application-protocol dce-rpc-portmap;
    protocol tcp;
    destination-port 135;
}
#
# DCE-RPC application on TCP sample UUID
# This application requires user to specify the UUID value
#
# application junos-dcerpc {
#     # application-protocol dce-rpc;
#     # protocol tcp;
#     #
#     # # UUID also needs to be defined as shown below
#     # UUID 11223344 22334455 33445566 44556677;
#     #
#     # }
#     #
#     # ms-exchange needs these 3 UUIDs
#     #
#     application junos-dcerpc-endpoint-mapper-service {
#         application-protocol dce-rpc;
#         protocol tcp;
#         uuid e1af8308-5d1f-11c9-91a4-08002b14a0fa;
#     }
#     application junos-dcerpc-msexchange-directory-rfr {
#         application-protocol dce-rpc;
#         protocol tcp;
#         uuid 1544f5e0-613c-11d1-93df-00c04fd7bd09;
#     }
#     application junos-dcerpc-msexchange-information-store {
#         application-protocol dce-rpc;
#         protocol tcp;
#         uuid a4f1db00-ca47-1067-b31f-00dd010662da;
#     }
# }
application junos-ssh {
```



```
    protocol tcp;
    destination-port 22;
}
application junos-telnet {
    protocol tcp;
    destination-port 23;
}
application junos-smtp {
    protocol tcp;
    destination-port 25;
}
application junos-dns-udp {
    protocol udp;
    destination-port 53;
}
application junos-dns-tcp {
    protocol tcp;
    destination-port 53;
}
application junos-tacacs {
    protocol tcp;
    destination-port 49;
}
# TACACS Database Service
application junos-tacacs-ds {
    protocol tcp;
    destination-port 65;
}
application junos-dhcp-client {
    protocol udp;
    destination-port 68;
}
application junos-dhcp-server {
    protocol udp;
    destination-port 67;
}
application junos-bootpc {
    protocol udp;
    destination-port 68;
}
application junos-bootps {
    protocol udp;
    destination-port 67;
}
application junos-finger {
    protocol tcp;
    destination-port 79;
}
application junos-http {
    protocol tcp;
    destination-port 80;
}
application junos-https {
    protocol tcp;
    destination-port 443;
}
```

```
application junos-pop3 {
    protocol tcp;
    destination-port 110;
}
application junos-ident {
    protocol tcp;
    destination-port 113;
}
application junos-nntp {
    protocol tcp;
    destination-port 119;
}
application junos-ntp {
    protocol udp;
    destination-port 123;
}
application junos-imap {
    protocol tcp;
    destination-port 143;
}
application junos-imaps {
    protocol tcp;
    destination-port 993;
}
application junos-bgp {
    protocol tcp;
    destination-port 179;
}
application junos-ldap {
    protocol tcp;
    destination-port 389;
}
application junos-snpp {
    protocol tcp;
    destination-port 444;
}
application junos-biff {
    protocol udp;
    destination-port 512;
}
# UNIX who
application junos-who {
    protocol udp;
    destination-port 513;
}
application junos-syslog {
    protocol udp;
    destination-port 514;
}
# line printer daemon, printer, spooler
application junos-printer {
    protocol tcp;
    destination-port 515;
}
# UNIX talk
application junos-talk-tcp {
```

```
    protocol tcp;
    destination-port 517;
}
application junos-talk-udp {
    protocol udp;
    destination-port 517;
}
application junos-ntalk {
    protocol udp;
    destination-port 518;
}
application junos-rip {
    protocol udp;
    destination-port 520;
}
# INA sanctioned RADIUS port numbers
application junos-radius {
    protocol udp;
    destination-port 1812;
}
application junos-radacct {
    protocol udp;
    destination-port 1813;
}
application junos-nfsd-tcp {
    protocol tcp;
    destination-port 2049;
}
application junos-nfsd-udp {
    protocol udp;
    destination-port 2049;
}
application junos-cvserver {
    protocol tcp;
    destination-port 2401;
}
#
# Label Distribution Protocol
#
application junos-ldp-tcp {
    protocol tcp;
    destination-port 646;
}
application junos-ldp-udp {
    protocol udp;
    destination-port 646;
}
#
# JUNOScript and JUNOScope management
#
application junos-xnm-ssl {
    protocol tcp;
    destination-port 3220;
}
application junos-xnm-clear-text {
    protocol tcp;
```

```
        destination-port 3221;
    }
    #
    # IPsec tunnel
    #
    application junos-ipsec-esp {
        protocol esp;
    }
    application junos-ike {
        protocol udp;
        destination-port 500;
    }
    #
    # 'junos-algs-outbound' defines a set of all applications
    # requiring an ALG. Useful for defining rule to the the public
    # internet allowing private network users to use all JUNOS OS
    # supported ALGs initiated from the private network.
    #
    # NOTE: the contents of this set might grow in future JUNOS OS versions.
    #
    application-set junos-algs-outbound {
        application junos-ftp;
        application junos-tftp;
        application junos-rpc-portmap-tcp;
        application junos-rpc-portmap-udp;
        application junos-snmp-get;
        application junos-snmp-get-next;
        application junos-snmp-response;
        application junos-snmp-trap;
        application junos-rexec;
        application junos-rlogin;
        application junos-rsh;
        application junos-rtsp;
        application junos-citrix-winframe;
        application junos-citrix-winframe-udp;
        application junos-sqlnet;
        application junos-h323;
        application junos-iiop-java;
        application junos-iiop-orbix;
        application junos-realaudio;
        application junos-traceroute;
        application junos-rpc-services-udp;
        application junos-rpc-services-tcp;
        application junos-icmp-all;
        application junos-netshow;
        application junos-netbios-name-udp;
        application junos-netbios-datagram;
        application junos-dcerpc-endpoint-mapper-service;
        application junos-dcerpc-msexchange-directory-rfr;
        application junos-dcerpc-msexchange-information-store;
    }
    #
    # 'junos-management-inbound' represents the group of applications
    # that might need access the router from public network for
    # for management purposes.
    #
```

```

# Set is intended for a UI to display management choices.
#
# NOTE: It is not recommended the user to use the entire set
# directly in a firewall rule and open up firewall to all
# of these applications. Also, the user should always
# specify the source and destination prefixes when using
# each application.
#
# NOTE: the contents of this set may grow in future JUNOS versions.
#
application-set junos-management-inbound {
    application junos-snmp-get;
    application junos-snmp-get-next;
    application junos-snmp-response;
    application junos-snmp-trap;
    application junos-ssh;
    application junos-telnet;
    application junos-http;
    application junos-https;
    application junos-xnm-ssl;
    application junos-xnm-clear-text;
}
#
# 'junos-routing-inbound' represents routing protocols that might
# need to access the router from public network.
#
# Set is intended for a UI to display routing involvement choices.
#
# NOTE: It is not recommended the user to use the entire set
# directly in a firewall rule and open up firewall to all
# of these applications. Also, the user should always
# specify the source and destination prefixes when using
# each application.
#
# NOTE: the contents of this set might grow in future JUNOS OS versions.
#
application-set junos-routing-inbound {
    application junos-bgp;
    application junos-rip;
    application junos-ldp-tcp;
    application junos-ldp-udp;
}
}

```

- Related Documentation**
- [Junos Default Groups on page 23](#)
 - [Configuring Application Sets on page 37](#)
 - [Configuring Application Protocol Properties on page 37](#)

Junos Default Groups

The Junos OS provides a default, hidden configuration group called **junos-defaults** that is automatically applied to the configuration of your router. The **junos-defaults** group contains preconfigured statements that contain predefined values for common

applications. Some of the statements must be referenced to take effect, such as applications like FTP or Telnet. Other statements are applied automatically, such as terminal settings. All of the preconfigured statements begin with the reserved name **junos-**.



NOTE: You can override the Junos default configuration values, but you cannot delete or edit them. If you delete a configuration, the defaults return when a new configuration is added.

You cannot use the **apply-groups** statement with the Junos defaults group.

To view the full set of available preset statements from the Junos default group, issue the **show groups junos-defaults** configuration mode command. The following example displays a partial list of Junos default groups that use application protocols (ALGs).

```
user@host# show groups junos-defaults
... output for other groups defined at the [edit groups junos-defaults] hierarchy level ...
applications {
  # File Transfer Protocol
  application junos-ftp {
    application-protocol ftp;
    protocol tcp;
    destination-port 21;
  }
  # Trivial File Transfer Protocol
  application junos-tftp {
    application-protocol tftp;
    protocol udp;
    destination-port 69;
  }
  # RPC port mapper on TCP
  application junos-rpc-portmap-tcp {
    application-protocol rpc-portmap;
    protocol tcp;
    destination-port 111;
  }
  # RPC port mapper on UDP
  application junos-rpc-portmap-udp {
    application-protocol rpc-portmap;
    protocol udp;
    destination-port 111;
  }
  # IP Protocol
  application junos-ip {
    application-protocol ip;
  }
  # remote exec
  application junos-rexec {
    application-protocol exec;
    protocol tcp;
    destination-port 512;
  }
  # remote login
```

```
application junos-rlogin {
    application-protocol login;
    protocol tcp;
    destination-port 513;
}
# remote shell
application junos-rsh {
    application-protocol shell;
    protocol tcp;
    destination-port 514;
}
# Real-Time Streaming Protocol
application junos-rtsp {
    application-protocol rtsp;
    protocol tcp;
    destination-port 554;
}
# Oracle SQL servers use this protocol to execute SQL commands
# from clients, load balance, use application-specific servers, and so on.
application junos-sqlnet {
    application-protocol sqlnet;
    protocol tcp;
    destination-port 1521;
}
# H.323 Protocol for audio/video conferencing
protocol tcp;
    destination-port 1720;
}
# Internet Inter-ORB Protocol is used for CORBA applications.
# The ORB protocol in Java virtual machine uses port 1975 as a default.
protocol tcp;
    destination-port 1975;
}
# Internet Inter-ORB Protocol is used for CORBA applications.
# ORBIX is a CORBA framework from Iona Technologies that uses
# port 3075 as a default.
protocol tcp;
    destination-port 3075;
}
# This was the original RealPlayer protocol.
# RTSP is more widely used by RealPlayer,
protocol tcp;
    destination-port 7070;
}
# Traceroute application
application junos-traceroute {
    application-protocol traceroute;
    protocol udp;
    destination-port 33435-33450;
    ttl-threshold 30;
}
# Traceroute application that stops at device supporting firewall
# (packets with ttl > 1 will be discarded).
application junos-traceroute-ttl-1 {
    application-protocol traceroute;
    protocol udp;
```

```
    destination-port 33435-33450;
    ttl-threshold 1;
}
# The full range of known RPC programs using UDP.
# Specific program numbers are assigned to certain applications.
application junos-rpc-services-udp {
    application-protocol rpc;
    protocol udp;
    rpc-program-number 100001-400000;
}
# The full range of known RPC programs using TCP.
# Specific program numbers are assigned to certain applications.
application junos-rpc-services-tcp {
    application-protocol rpc;
    protocol tcp;
    rpc-program-number 100001-400000;
}
# All ICMP traffic
# This can be made more restrictive by specifying ICMP type and code.
application junos-icmp-all {
    application-protocol icmp;
}
# ICMP ping; the echo reply is allowed upon return.
application junos-icmp-ping {
    application-protocol icmp;
    icmp-type echo-request;
}
# Protocol used by Windows Media Server and Windows Media Player
application junos-netshow {
    application-protocol netshow;
    protocol tcp;
    destination-port 1755;
}
# NetBIOS, the networking protocol used on Windows networks;
# includes name service port, both UDP and TCP.
application junos-netbios-name-udp {
    application-protocol netbios;
    protocol udp;
    destination-port 137;
}
application junos-netbios-name-tcp {
    protocol tcp;
    destination-port 137;
}
# NetBIOS, the networking protocol used on Windows networks;
# includes datagram service port.
application junos-netbios-datagram {
    application-protocol netbios;
    protocol udp;
    destination-port 138;
}
# NetBIOS, the networking protocol used on Windows networks;
# includes session service port.
application junos-netbios-session {
    protocol tcp;
    destination-port 139;
```



```
}
# DCE-RPC port mapper on TCP
application junos-dce-rpc-portmap {
    application-protocol dce-rpc-portmap;
    protocol tcp;
    destination-port 135;
}
# MS Exchange requires these three UUID values.
application junos-dcerpc-endpoint-mapper-service {
    application-protocol dce-rpc;
    protocol tcp;
    uuid e1af8308-5d1f-11c9-91a4-08002b14a0fa;
}
application junos-ssh {
    protocol tcp;
    destination-port 22;
}
application junos-telnet {
    protocol tcp;
    destination-port 23;
}
application junos-smtp {
    protocol tcp;
    destination-port 25;
}
application junos-dns-udp {
    protocol udp;
    destination-port 53;
}
application junos-dns-tcp {
    protocol tcp;
    destination-port 53;
}
application junos-tacacs {
    protocol tcp;
    destination-port 49;
}
# TACACS Database Service
application junos-tacacs-ds {
    protocol tcp;
    destination-port 65;
}
application junos-dhcp-client {
    protocol udp;
    destination-port 68;
}
application junos-dhcp-server {
    protocol udp;
    destination-port 67;
}
application junos-bootpc {
    protocol udp;
    destination-port 68;
}
application junos-bootps {
    protocol udp;
```

```
    destination-port 67;
}
application junos-http {
    protocol tcp;
    destination-port 80;
}
application junos-https {
    protocol tcp;
    destination-port 443;
}
# "junos-algs-outbound" defines a set of all applications
# requiring an ALG. Useful for defining a rule for an untrusted
# network to allow trusted network users to use all the
# Junos-supported ALGs initiated from the trusted network.
application-set junos-algs-outbound {
    application junos-ftp;
    application junos-tftp;
    application junos-rpc-portmap-tcp;
    application junos-rpc-portmap-udp;
    application junos-snmp-get;
    application junos-snmp-get-next;
    application junos-snmp-response;
    application junos-snmp-trap;
    application junos-rexec;
    application junos-rlogin;
    application junos-rsh;
    application junos-rtsp;
    application junos-sqlnet;
    application junos-traceroute;
    application junos-rpc-services-udp;
    application junos-rpc-services-tcp;
    application junos-icmp-all;
    application junos-netshow;
    application junos-netbios-name-udp;
    application junos-netbios-datagram;
    application junos-dce-rpc-portmap;
    application junos-dcerpc-msexchange-directory-rfr;
    application junos-dcerpc-msexchange-information-store;
    application junos-dcerpc-msexchange-directory-nsp;
}
# "junos-management-inbound" represents the group of applications
# that might need access to the trusted network from the untrusted
# network for management purposes.
# The set is intended for a UI to display management choices.
# NOTE: It is not recommended that you use the entire set directly in
# a firewall rule and open up firewall to all of these
# applications. Also, you should always specify the source
# and destination prefixes when using each application.
application-set junos-management-inbound {
    application junos-snmp-get;
    application junos-snmp-get-next;
    application junos-snmp-response;
    application junos-snmp-trap;
    application junos-ssh;
    application junos-telnet;
    application junos-http;
```

```

    application junos-https;
    application junos-xnm-ssl;
    application junos-xnm-clear-text;
    application junos-icmp-ping;
    application junos-traceroute-ttl-1;
  }
}
}
}

```

To reference statements available from the **junos-defaults** group, include the selected **junos-default-name** statement at the applicable hierarchy level. To configure application protocols, see [“Configuring Application Protocol Properties” on page 37](#); for details about a specific protocol, see [“ALG Descriptions” on page 3](#).

Examples: Referencing the Preset Statement from the Junos Default Group

The following example is a preset statement from the Junos default groups that is available for FTP in a stateful firewall:

```

[edit]
groups {
  junos-defaults {
    applications {
      application junos-ftp { # Use FTP default configuration
        application-protocol ftp;
        protocol tcp;
        destination-port 21;
      }
    }
  }
}

```

To reference a preset Junos default statement from the Junos default groups, include the **junos-default-name** statement at the applicable hierarchy level. For example, to reference the Junos default statement for FTP in a stateful firewall, include the **junos-ftp** statement at the **[edit services stateful-firewall rule rule-name term term-name from applications]** hierarchy level.

```

[edit]
services {
  stateful-firewall {
    rule my-rule {
      term my-term {
        from {
          applications junos-ftp; #Reference predefined statement, junos-ftp,
        }
      }
    }
  }
}

```

The following example shows configuration of the default Junos IP ALG:

```

[edit]
services {
  stateful-firewall {

```

```
rule r1 {
  match-direction input;
  term t1 {
    from {
      applications junos-ip;
    }
    then {
      accept;
      syslog;
    }
  }
}
```

If you configure the IP ALG in the stateful firewall rule, it is matched by any IP traffic, but if there is any other more specific application that matches the same traffic, the IP ALG will not be matched. For example, in the following configuration, both the ICMP ALG and the IP ALG are configured, but traffic is matched for ICMP packets, because it is the more specific match.

```
[edit]
services {
  stateful-firewall {
    rule r1 {
      match-direction input;
      term t1 {
        from {
          applications [ junos-ip junos-icmp-all ];
        }
        then {
          accept;
          syslog;
        }
      }
    }
  }
}
```

Related Documentation

- [ALG Descriptions on page 3](#)
- [Configuring Application Sets on page 37](#)
- [Configuring Application Protocol Properties on page 37](#)

ALGs Available by Default for Junos OS Address Aware NAT

The following application-level gateways (ALGs) listed in [Table 6 on page 31](#) are supported for NAT processing on the listed platforms.

To view the implementation details (port, protocol, and so on) for these Junos OS default applications, locate the Junos OS Default ALG Name in the table and then look up the listed name in the **groups**. For example, for details about TFTP, look up **junos-tftp** as shown.



TIP: The Junos OS provides the `junos-alg`, which enables other ALGs to function by handling ALG registrations, causing slow path packets to flow through registered ALGs, and transferring ALG events to the ALG plug-ins. The `junos-alg` ALG is automatically available on the MS-MPC and MS-MIC platforms and does not require further configuration.

```
user@host# show groups junos-defaults applications application junos-tftp
application-protocol tftp;
protocol udp;
destination-port 69;
```

Table 6: ALGs Available by Default

ALG	MS-DPC	MS-MPC, MS-MIC	Junos OS Default ALG Name
Basic TCP ALG	yes	yes	NOTE: Specific Junos ALGs are not supported. However, a feature called TCP tracker, available by default, performs segment ordering and retransmit and connection tracking, validations for TCP connections.
Basic UDP ALG	yes	yes	NOTE: TCP tracker performs limited integrity and validation checks for UDP.
BOOTP	yes	no	<ul style="list-style-type: none"> <code>junos-bootpc</code> <code>junos-bootps</code>
DCE RPC Services	yes	yes	<ul style="list-style-type: none"> <code>junos-dce-rpc-portmap</code> <code>junos-dcerpc-endpoint-mapper-service</code> <code>junos-dcerpc-msexchange-directory-nsp</code> <code>junos-dcerpc-msexchange-directory-rfr</code> <code>junos-dcerpc-msexchange-information-store</code>
DNS	yes	yes	<ul style="list-style-type: none"> <code>junos-dns-tcp</code> <code>junos-dns-udp</code>
FTP	yes	yes	<ul style="list-style-type: none"> <code>junos-ftp</code>
H323	yes	no	<ul style="list-style-type: none"> <code>junos-h323</code>
ICMP	yes	yes NOTE: ICMP messages are handled by default, but PING ALG support is not provided.	<ul style="list-style-type: none"> <code>junos-icmp-all</code> <code>junos-icmp-ping</code>
IIOIP	yes	no	<ul style="list-style-type: none"> <code>junos-iiop-java</code> <code>junos-iiop-orbix</code>

Table 6: ALGs Available by Default (*continued*)

ALG	MS-DPC	MS-MPC, MS-MIC	Junos OS Default ALG Name
IP	yes	The TCP tracker, available by default on these platforms, performs limited integrity and validation checks.	<ul style="list-style-type: none"> • junos-ip
NETBIOS	yes	no	<ul style="list-style-type: none"> • junos-netbios-datagram • junos-netbios-name-tcp • junos-netbios-name-udp • junos-netbios-session
NETSHOW	yes	no	<ul style="list-style-type: none"> • junos-netshow
PPTP	yes	yes	<ul style="list-style-type: none"> • junos-pptp
REALAUDIO	yes	no	<ul style="list-style-type: none"> • junos-realaudio
Sun RPC and RPC Port Map Services	yes	yes	<ul style="list-style-type: none"> • junos-rpc-portmap-tcp • junos-rpc-portmap-udp
RTSP	yes	yes	<ul style="list-style-type: none"> • junos-rtsp
SIP	yes	Yes	<ul style="list-style-type: none"> • junos-sip
SNMP	yes	No	<ul style="list-style-type: none"> • junos-snmp-get • junos-snmp-get-next • junos-snmp-response junos-snmp-trap
SQLNET	yes	yes	<ul style="list-style-type: none"> • junos-sqlnet
TFTP	yes	yes	<ul style="list-style-type: none"> • junos-tftp
Traceroute	yes	no	<ul style="list-style-type: none"> • junos-traceroute
Unix Remote Shell Service	yes	Yes	<ul style="list-style-type: none"> • junos-rsh
WINFrame	yes	No	<ul style="list-style-type: none"> • junos-citrix-winframe • junos-citrix-winframe-udp
TALK-UDP	No	Yes	<ul style="list-style-type: none"> • junos-talk-udp
MS RPC	No	Yes	<ul style="list-style-type: none"> • junos-rpc-portmap-tcp • junos-rpc-portmap-udp • junos-rpc-services-tcp • junos-rpc-services-udp

Related Documentation • *Network Address Translation Configuration Overview*

PART 2

Configuration

- [Configuration Tasks on page 37](#)
- [Example on page 55](#)
- [Verification on page 57](#)
- [Configuration Statements on page 65](#)

CHAPTER 2

Configuration Tasks

- [Configuring Application Sets on page 37](#)
- [Configuring Application Protocol Properties on page 37](#)

Configuring Application Sets

You can group the applications you have defined into a named object by including the **application-set** statement at the **[edit applications]** hierarchy level with an **application** statement for each application:

```
[edit applications]
  application-set application-set-name {
    application application;
  }
```

For an example of a typical application set, see “Examples: Configuring Application Protocols” on page 55.

Related Documentation

- [ALG Descriptions on page 3](#)
- [Configuring Application Protocol Properties on page 37](#)
- [Examples: Configuring Application Protocols on page 55](#)
- [Verifying the Output of ALG Sessions on page 57](#)

Configuring Application Protocol Properties

To configure application properties, include the **application** statement at the **[edit applications]** hierarchy level:

```
[edit applications]
  application application-name {
    application-protocol protocol-name;
    destination-port port-number;
    icmp-code value;
    icmp-type value;
    inactivity-timeout value;
    protocol type;
    rpc-program-number number;
    snmp-command command;
```

```

source-port port-number;
ttl-threshold value;
uuid hex-value;
}

```

You can group application objects by configuring the **application-set** statement; for more information, see [“Configuring Application Sets” on page 37](#).

This section includes the following tasks for configuring applications:

- [Configuring an Application Protocol on page 38](#)
- [Configuring the Network Protocol on page 40](#)
- [Configuring the ICMP Code and Type on page 41](#)
- [Configuring Source and Destination Ports on page 43](#)
- [Configuring the Inactivity Timeout Period on page 46](#)
- [Configuring SIP on page 46](#)
- [Configuring an SNMP Command for Packet Matching on page 53](#)
- [Configuring an RPC Program Number on page 53](#)
- [Configuring the TTL Threshold on page 54](#)
- [Configuring a Universal Unique Identifier on page 54](#)

Configuring an Application Protocol

The **application-protocol** statement allows you to specify which of the supported application protocols (ALGs) to configure and include in an application set for service processing. To configure application protocols, include the **application-protocol** statement at the **[edit applications application application-name]** hierarchy level:

```

[edit applications application application-name]
application-protocol protocol-name;

```

[Table 7 on page 38](#) shows the list of supported protocols. For more information about specific protocols, see [“ALG Descriptions” on page 3](#).

Table 7: Application Protocols Supported by Services Interfaces

Protocol Name	CLI Value	Comments
Bootstrap protocol (BOOTP)	bootp	Supports BOOTP and dynamic host configuration protocol (DHCP).
Distributed Computing Environment (DCE) remote procedure call (RPC)	dce-rpc	Requires the protocol statement to have the value udp or tcp . Requires a uuid value. You cannot specify destination-port or source-port values.
DCE RPC portmap	dce-rpc-portmap	Requires the protocol statement to have the value udp or tcp . Requires a destination-port value.
Domain Name System (DNS)	dns	Requires the protocol statement to have the value udp . This application protocol closes the DNS flow as soon as the DNS response is received.
Exec	exec	Requires the protocol statement to have the value tcp or to be unspecified. Requires a destination-port value.

Table 7: Application Protocols Supported by Services Interfaces (*continued*)

Protocol Name	CLI Value	Comments
FTP	ftp	Requires the protocol statement to have the value tcp or to be unspecified. Requires a destination-port value.
H.323	h323	—
Internet Control Message Protocol (ICMP)	icmp	Requires the protocol statement to have the value icmp or to be unspecified.
Internet Inter-ORB Protocol	iiop	—
IP	ip	—
Login	login	—
NetBIOS	netbios	Requires the protocol statement to have the value udp or to be unspecified. Requires a destination-port value.
NetShow	netshow	Requires the protocol statement to have the value tcp or to be unspecified. Requires a destination-port value.
Point-to-Point Tunneling Protocol	pptp	—
RealAudio	realaudio	—
Real-Time Streaming Protocol (RTSP)	rtsp	Requires the protocol statement to have the value tcp or to be unspecified. Requires a destination-port value.
RPC User Datagram Protocol (UDP) or TCP	rpc	Requires the protocol statement to have the value udp or tcp . Requires a rpc-program-number value. You cannot specify destination-port or source-port values.
RPC port mapping	rpc-portmap	Requires the protocol statement to have the value udp or tcp . Requires a destination-port value.
Shell	shell	Requires the protocol statement to have the value tcp or to be unspecified. Requires a destination-port value.
Session Initiation Protocol	sip	—
SNMP	snmp	Requires the protocol statement to have the value udp or to be unspecified. Requires a destination-port value.
SQLNet	sqlnet	Requires the protocol statement to have the value tcp or to be unspecified. Requires a destination-port or source-port value.
Talk Program	talk	

Table 7: Application Protocols Supported by Services Interfaces (*continued*)

Protocol Name	CLI Value	Comments
Trace route	traceroute	Requires the protocol statement to have the value udp or to be unspecified. Requires a destination-port value.
Trivial FTP (TFTP)	tftp	Requires the protocol statement to have the value udp or to be unspecified. Requires a destination-port value.
WinFrame	winframe	—



NOTE: You can configure application-level gateways (ALGs) for ICMP and trace route under stateful firewall, NAT, or CoS rules when twice NAT is configured in the same service set. These ALGs cannot be applied to flows created by the Packet Gateway Controller Protocol (PGCP). Twice NAT does not support any other ALGs. NAT applies only the IP address and TCP or UDP headers, but not the payload.

For more information about configuring twice NAT, see *Junos Address Aware Carrier Grade NAT and IPv6 Feature Guide*.

Configuring the Network Protocol

The **protocol** statement allows you to specify which of the supported network protocols to match in an application definition. To configure network protocols, include the **protocol** statement at the **[edit applications application *application-name*]** hierarchy level:

```
[edit applications application application-name]
  protocol type;
```

You specify the protocol type as a numeric value; for the more commonly used protocols, text names are also supported in the command-line interface (CLI). [Table 8 on page 40](#) shows the list of the supported protocols.

Table 8: Network Protocols Supported by Services Interfaces

Network Protocol Type	CLI Value	Comments
IP Security (IPsec) authentication header (AH)	ah	—
External Gateway Protocol (EGP)	egp	—
IPsec Encapsulating Security Payload (ESP)	esp	—
Generic routing encapsulation (GR)	gre	—
ICMP	icmp	Requires an application-protocol value of icmp .

Table 8: Network Protocols Supported by Services Interfaces (*continued*)

Network Protocol Type	CLI Value	Comments
ICMPv6	icmp6	Requires an application-protocol value of icmp .
Internet Group Management Protocol (IGMP)	igmp	—
IP in IP	ipip	—
OSPF	ospf	—
Protocol Independent Multicast (PIM)	pim	—
Resource Reservation Protocol (RSVP)	rsvp	—
TCP	tcp	Requires a destination-port or source-port value unless you specify application-protocol rcp or dce-rcp .
UDP	udp	Requires a destination-port or source-port value unless you specify application-protocol rcp or dce-rcp .

For a complete list of possible numeric values, see RFC 1700, *Assigned Numbers (for the Internet Protocol Suite)*.



NOTE: IP version 6 (IPv6) is not supported as a network protocol in application definitions.

By default, the twice NAT feature can affect IP, TCP, and UDP headers embedded in the payload of ICMP error messages. You can include the **protocol tcp** and **protocol udp** statements with the application statement for twice NAT configurations. For more information about configuring twice NAT, see *Junos Address Aware Carrier Grade NAT and IPv6 Feature Guide*.

Configuring the ICMP Code and Type

The ICMP code and type provide additional specification, in conjunction with the network protocol, for packet matching in an application definition. To configure ICMP settings, include the **icmp-code** and **icmp-type** statements at the **[edit applications application application-name]** hierarchy level:

```
[edit applications application application-name]
  icmp-code value;
  icmp-type value;
```

You can include only one ICMP code and type value. The **application-protocol** statement must have the value **icmp**. Table 9 on page 42 shows the list of supported ICMP values.

Table 9: ICMP Codes and Types Supported by Services Interfaces

CLI Statement	Description
icmp-code	<p>This value or keyword provides more specific information than icmp-type. Because the value's meaning depends upon the associated icmp-type value, you must specify icmp-type along with icmp-code. For more information, see the <i>Routing Policy Feature Guide for Routing Devices</i>.</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> <p>parameter-problem: ip-header-bad (0), required-option-missing (1)</p> <p>redirect: redirect-for-host (1), redirect-for-network (0), redirect-for-tos-and-host (3), redirect-for-tos-and-net (2)</p> <p>time-exceeded: ttl-eq-zero-during-reassembly (1), ttl-eq-zero-during-transit (0)</p> <p>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)</p>
icmp-type	<p>Normally, you specify this match in conjunction with the protocol match statement to determine which protocol is being used on the port. For more information, see the <i>Routing Policy Feature Guide for Routing Devices</i>.</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>



NOTE: If you configure an interface with an input firewall filter that includes a reject action and with a service set that includes stateful firewall rules, the router executes the input firewall filter before the stateful firewall rules are run on the packet. As a result, when the Packet Forwarding Engine sends an ICMP error message out through the interface, the stateful firewall rules might drop the packet because it was not seen in the input direction.

Possible workarounds are to include a forwarding-table filter to perform the reject action, because this type of filter is executed after the stateful firewall in the input direction, or to include an output service filter to prevent the locally generated ICMP packets from going to the stateful firewall service.

Configuring Source and Destination Ports

The TCP or UDP source and destination port provide additional specification, in conjunction with the network protocol, for packet matching in an application definition. To configure ports, include the **destination-port** and **source-port** statements at the **[edit applications application application-name]** hierarchy level:

```
[edit applications application application-name]
destination-port value;
source-port value;
```

You must define one source or destination port. Normally, you specify this match in conjunction with the **protocol** match statement to determine which protocol is being used on the port; for constraints, see [Table 7 on page 38](#).

You can specify either a numeric value or one of the text synonyms listed in [Table 10 on page 43](#).

Table 10: Port Names Supported by Services Interfaces

Port Name	Corresponding Port Number
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

Table 10: Port Names Supported by Services Interfaces (*continued*)

Port Name	Corresponding Port Number
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
nntp	119
ntalk	518
ntp	123
pop3	110
pptp	1723

Table 10: Port Names Supported by Services Interfaces (*continued*)

Port Name	Corresponding Port Number
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-ds	65
talk	517
telnet	23
tftp	69
timed	525
who	513
xmcp	177
zephyr-clt	2103
zephyr-hm	2104

For more information about matching criteria, see the *Routing Policy Feature Guide for Routing Devices*.

Configuring the Inactivity Timeout Period

You can specify a timeout period for application inactivity. If the software has not detected any activity during the duration, the flow becomes invalid when the timer expires. To configure a timeout period, include the **inactivity-timeout** statement at the **[edit applications application application-name]** hierarchy level:

```
[edit applications application application-name]
inactivity-timeout seconds;
```

The default value is 30 seconds. The value you configure for an application overrides any global value configured at the **[edit interfaces interface-name service-options]** hierarchy level; for more information, see *Configuring Default Timeout Settings for Services Interfaces*.

Configuring SIP

The Session Initiation Protocol (SIP) is a generalized protocol for communication between endpoints involved in Internet services such as telephony, fax, video conferencing, instant messaging, and file exchange.

The Junos OS provides ALG services in accordance with the standard described in RFC 3261, *SIP: Session Initiation Protocol*. SIP flows under the Junos OS are as described in RFC 3665, *Session Initiation Protocol (SIP) Basic Call Flow Examples*.



NOTE: Before implementing the Junos OS SIP ALG, you should be familiar with certain limitations, discussed in [“Junos OS SIP ALG Limitations” on page 52](#)

The use of NAT in conjunction with the SIP ALG results in changes in SIP header fields due to address translation. For an explanation of these translations, refer to [“SIP ALG Interaction with Network Address Translation” on page 47](#).

To implement SIP on adaptive services interfaces, you configure the **application-protocol** statement at the **[edit applications application application-name]** hierarchy level with the value **sip**. For more information about this statement, see [“Configuring an Application Protocol” on page 38](#). In addition, there are two other statements you can configure to modify how SIP is implemented:

- You can enable the router to accept any incoming SIP calls for the endpoint devices that are behind the NAT firewall. When a device behind the firewall registers with the proxy that is outside the firewall, the AS or Multiservices PIC maintains the registration state. When the **learn-sip-register** statement is enabled, the router can use this information to accept inbound calls. If this statement is not configured, no inbound calls are accepted; only the devices behind the firewall can call devices outside the firewall.

To configure SIP registration, include the **learn-sip-register** statement at the **[edit applications application application-name]** hierarchy level:

```
[edit applications application application-name]
```

```
learn-sip-register;
```

You can also manually inspect the SIP register by issuing the **show services stateful-firewall sip-register** command; for more information, see the *Junos OS System Basics and Services Command Reference*.

- You can specify a timeout period for the duration of SIP calls that are placed on hold. When a call is put on hold, there is no activity and flows might time out after the configured **inactivity-timeout** period expires, resulting in call state teardown. To avoid this, when a call is put on hold, the flow timer is reset to the **sip-call-hold-timeout** cycle to preserve the call state and flows for longer than the **inactivity-timeout** period.

To configure a timeout period, include the **sip-call-hold-timeout** statement at the **[edit applications application *application-name*]** hierarchy level:

```
[edit applications application application-name]
sip-call-hold-timeout seconds;
```

The default value is 7200 seconds and the range is from 0 through 36,000 seconds (10 hours).

SIP ALG Interaction with Network Address Translation

The Network Address Translation (NAT) protocol enables multiple hosts in a private subnet to share a single public IP address to access the Internet. For outgoing traffic, NAT replaces the private IP address of the host in the private subnet with the public IP address. For incoming traffic, the public IP address is converted back into the private address, and the message is routed to the appropriate host in the private subnet.

Using NAT with the Session Initiation Protocol (SIP) service is more complicated because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. When using NAT with the SIP service, the SIP headers contain information about the caller and the receiver, and the device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media. The device translates SDP information for allocating resources to send and receive the media.

How IP addresses and port numbers in SIP messages are replaced depends on the direction of the message. For an outgoing message, the private IP address and port number of the client are replaced with the public IP address and port number of the Juniper Networks firewall. For an incoming message, the public address of the firewall is replaced with the private address of the client.

When an INVITE message is sent out across the firewall, the SIP Application Layer Gateway (ALG) collects information from the message header into a call table, which it uses to forward subsequent messages to the correct endpoint. When a new message arrives, for example an ACK or 200 OK, the ALG compares the "From:", "To:", and "Call-ID:" fields against the call table to identify the call context of the message. If a new INVITE message arrives that matches the existing call, the ALG processes it as a REINVITE.

When a message containing SDP information arrives, the ALG allocates ports and creates a NAT mapping between them and the ports in the SDP. Because the SDP requires sequential ports for the Real-Time Transport Protocol (RTP) and Real-Time Control

Protocol (RTCP) channels, the ALG provides consecutive even-odd ports. If it is unable to find a pair of ports, it discards the SIP message.

This topic contains the following sections:

- [Outgoing Calls on page 48](#)
- [Incoming Calls on page 48](#)
- [Forwarded Calls on page 49](#)
- [Call Termination on page 49](#)
- [Call Re-INVITE Messages on page 49](#)
- [Call Session Timers on page 49](#)
- [Call Cancellation on page 50](#)
- [Forking on page 50](#)
- [SIP Messages on page 50](#)
- [SIP Headers on page 50](#)
- [SIP Body on page 52](#)

Outgoing Calls

When a SIP call is initiated with a SIP request message from the internal to the external network, NAT replaces the IP addresses and port numbers in the SDP and binds the IP addresses and port numbers to the Juniper Networks firewall. Via, Contact, Route, and Record-Route SIP header fields, if present, are also bound to the firewall IP address. The ALG stores these mappings for use in retransmissions and for SIP response messages.

The SIP ALG then opens pinholes in the firewall to allow media through the device on the dynamically assigned ports negotiated based on information in the SDP and the Via, Contact, and Record-Route header fields. The pinholes also allow incoming packets to reach the Contact, Via, and Record-Route IP addresses and ports. When processing return traffic, the ALG inserts the original Contact, Via, Route, and Record-Route SIP fields back into packets.

Incoming Calls

Incoming calls are initiated from the public network to public static NAT addresses or to interface IP addresses on the device. Static NATs are statically configured IP addresses that point to internal hosts; interface IP addresses are dynamically recorded by the ALG as it monitors REGISTER messages sent by internal hosts to the SIP registrar. When the device receives an incoming SIP packet, it sets up a session and forwards the payload of the packet to the SIP ALG.

The ALG examines the SIP request message (initially an INVITE) and, based on information in the SDP, opens gates for outgoing media. When a 200 OK response message arrives, the SIP ALG performs NAT on the IP addresses and ports and opens pinholes in the outbound direction. (The opened gates have a short time-to-live, and they time out if a 200 OK response message is not received quickly.)

When a 200 OK response arrives, the SIP proxy examines the SDP information and reads the IP addresses and port numbers for each media session. The SIP ALG on the device

performs NAT on the addresses and port numbers, opens pinholes for outbound traffic, and refreshes the timeout for gates in the inbound direction.

When the ACK arrives for the 200 OK, it also passes through the SIP ALG. If the message contains SDP information, the SIP ALG ensures that the IP addresses and port numbers are not changed from the previous INVITE—if they are, the ALG deletes old pinholes and creates new pinholes to allow media to pass through. The ALG also monitors the Via, Contact, and Record-Route SIP fields and opens new pinholes if it determines that these fields have changed.

Forwarded Calls

A forwarded call is when, for example, user A outside the network calls user B inside the network, and user B forwards the call to user C outside the network. The SIP ALG processes the INVITE from user A as a normal incoming call. But when the ALG examines the forwarded call from B to C outside the network and notices that B and C are reached using the same interface, it does not open pinholes in the firewall, because media will flow directly between user A and user C.

Call Termination

The BYE message terminates a call. When the device receives a BYE message, it translates the header fields just as it does for any other message. But because a BYE message must be acknowledged by the receiver with a 200 OK, the ALG delays call teardown for five seconds to allow time for transmission of the 200 OK.

Call Re-INVITE Messages

Re-INVITE messages add new media sessions to a call and remove existing media sessions. When new media sessions are added to a call, new pinholes are opened in the firewall and new address bindings are created. The process is identical to the original call setup. When one or more media sessions are removed from a call, pinholes are closed and bindings released just as with a BYE message.

Call Session Timers

The SIP ALG uses the Session-Expires value to time out a session if a Re-INVITE or UPDATE message is not received. The ALG gets the Session-Expires value, if present, from the 200 OK response to the INVITE and uses this value for signaling timeout. If the ALG receives another INVITE before the session times out, it resets all timeout values to this new INVITE or to default values, and the process is repeated.

As a precautionary measure, the SIP ALG uses hard timeout values to set the maximum amount of time a call can exist. This ensures that the device is protected should one of the following events occur:

- End systems crash during a call and a BYE message is not received.
- Malicious users never send a BYE in an attempt to attack a SIP ALG.
- Poor implementations of SIP proxy fail to process Record-Route and never send a BYE message.
- Network failures prevent a BYE message from being received.

Call Cancellation

Either party can cancel a call by sending a CANCEL message. Upon receiving a CANCEL message, the SIP ALG closes pinholes through the firewall—if any have been opened—and releases address bindings. Before releasing the resources, the ALG delays the control channel age-out for approximately five seconds to allow time for the final 200 OK to pass through. The call is terminated when the five second timeout expires, regardless of whether a 487 or non-200 response arrives.

Forking

Forking enables a SIP proxy to send a single INVITE message to multiple destinations simultaneously. When the multiple 200 OK response messages arrive for the single call, the SIP ALG parses but updates call information with the first 200 OK messages it receives.

SIP Messages

The SIP message format consists of a SIP header section and the SIP body. In request messages, the first line of the header section is the request line, which includes the method type, request-URI, and protocol version. In response messages, the first line is the status line, which contains a status code. SIP headers contain IP addresses and port numbers used for signaling. The SIP body, separated from the header section by a blank line, is reserved for session description information, which is optional. Junos OS currently supports the SDP only. The SIP body contains IP addresses and port numbers used to transport the media.

SIP Headers

In the following sample SIP request message, NAT replaces the IP addresses in the header fields to hide them from the outside network.

```
INVITE bob@10.150.20.5 SIP/2.0
Via: SIP/2.0/UDP 10.150.20.3:5434
From: alice@10.150.20.3
To: bob@10.150.20.5
Call-ID: a12abcde@10.150.20.3
Contact: alice@10.150.20.3:5434
Route: <sip:netscreen@10.150.20.3:5060>
Record-Route: <sip:netscreen@10.150.20.3:5060>
```

How IP address translation is performed depends on the type and direction of the message. A message can be any of the following:

- Inbound request
- Outbound response
- Outbound request
- Inbound response

[Table 11 on page 51](#) shows how NAT is performed in each of these cases. Note that for several of the header fields the ALG determine more than just whether the messages comes from inside or outside the network. It must also determine what client initiated the call, and whether the message is a request or response.

Table 11: Requesting Messages with NAT Table

Inbound Request (from public to private)	To:	Replace domain with local address
	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	Replace ALG address with local address
	Contact:	None
	Record-Route:	None
	Route:	None
Outbound Response (from private to public)	To:	Replace ALG address with local address
	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	N/A
	Contact:	Replace local address with ALG address
	Record-Route:	Replace local address with ALG address
	Route:	None
Outbound Request (from private to public)	To:	None
	From:	Replace local address with ALG address
	Call-ID:	None
	Via:	Replace local address with ALG address
	Request-URI:	None
	Contact:	Replace local address with ALG address
	Record-Route:	Replace local address with ALG address
	Route:	Replace ALG address with local address

Table 11: Requesting Messages with NAT Table (*continued*)

Outbound Response (from public to private)	To:	None
	From:	Replace ALG address with local address
	Call-ID:	None
	Via:	Replace ALG address with local address
	Request-URI:	N/A
	Contact:	None
	Record-Route:	Replace ALG address with local address
	Route:	Replace ALG address with local address

SIP Body

The SDP information in the SIP body includes IP addresses the ALG uses to create channels for the media stream. Translation of the SDP section also allocates resources, that is, port numbers to send and receive the media.

The following excerpt from a sample SDP section shows the fields that are translated for resource allocation.

```
o=user 2344234 55234434 IN IP4 10.150.20.3
c=IN IP4 10.150.20.3
m=audio 43249 RTP/AVP 0
```

SIP messages can contain more than one media stream. The concept is similar to attaching multiple files to an e-mail message. For example, an INVITE message sent from a SIP client to a SIP server might have the following fields:

```
c=IN IP4 10.123.33.4
m=audio 33445 RTP/AVP 0
c=IN IP4 10.123.33.4
m=audio 33447 RTP/AVP 0
c=IN IP4 10.123.33.4
m=audio 33449 RTP/AVP 0
```

Junos OS supports up to 6 SDP channels negotiated for each direction, for a total of 12 channels per call. For more information, see *SDP Session Descriptions*.

Junos OS SIP ALG Limitations

The following limitations apply to configuration of the SIP ALG:

- Only the methods described in RFC 3261 are supported.
- Only SIP version 2 is supported.
- TCP is not supported as a transport mechanism for signaling messages.

- *Do not configure the SIP ALG when using STUN.* If clients use STUN/TURN to detect the firewall or NAT devices between the caller and responder or proxy, the client attempts to best-guess the NAT device behavior and act accordingly to place the call.
- Do not use the endpoint-independent mapping NAT pool option in conjunction with the SIP ALG. Errors will result.
- IPv6 signaling data is not supported.
- Authentication is not supported.
- Encrypted messages are not supported.
- SIP fragmentation is not supported.
- The maximum UDP packet size containing a SIP message is assumed to be 4 KB. SIP messages larger than this are not supported.
- The maximum number of media channels in a SIP message is assumed to be six.
- Fully qualified domain names (FQDNs) are not supported in critical fields.
- QoS is not supported.
- High availability is not supported, except for warm standby.
- A timeout setting of never is not supported on SIP or NAT.
- Multicast (forking proxy) is not supported.

Configuring an SNMP Command for Packet Matching

You can specify an SNMP command setting for packet matching. To configure SNMP, include the **snmp-command** statement at the **[edit applications application application-name]** hierarchy level:

```
[edit applications application application-name]
snmp-command value;
```

The supported values are **get**, **get-next**, **set**, and **trap**. You can configure only one value for matching. The **application-protocol** statement at the **[edit applications application application-name]** hierarchy level must have the value **snmp**. For information about specifying the application protocol, see [“Configuring an Application Protocol” on page 38](#).

Configuring an RPC Program Number

You can specify an RPC program number for packet matching. To configure an RPC program number, include the **rpc-program-number** statement at the **[edit applications application application-name]** hierarchy level:

```
[edit applications application application-name]
rpc-program-number number;
```

The range of values used for DCE or RPC is from 100,000 through 400,000. The **application-protocol** statement at the **[edit applications application application-name]** hierarchy level must have the value **rpc**. For information about specifying the application protocol, see [“Configuring an Application Protocol” on page 38](#).

Configuring the TTL Threshold

You can specify a trace route time-to-live (TTL) threshold value, which controls the acceptable level of network penetration for trace routing. To configure a TTL value, include the **ttl-threshold** statement at the **[edit applications application application-name]** hierarchy level:

```
[edit applications application application-name]
ttl-threshold value;
```

The **application-protocol** statement at the **[edit applications application application-name]** hierarchy level must have the value **traceroute**. For information about specifying the application protocol, see [“Configuring an Application Protocol” on page 38](#).

Configuring a Universal Unique Identifier

You can specify a Universal Unique Identifier (UUID) for DCE RPC objects. To configure a UUID value, include the **uuid** statement at the **[edit applications application application-name]** hierarchy level:

```
[edit applications application application-name]
uuid hex-value;
```

The **uuid** value is in hexadecimal notation. The **application-protocol** statement at the **[edit applications application application-name]** hierarchy level must have the value **dce-rpc**. For information about specifying the application protocol, see [“Configuring an Application Protocol” on page 38](#). For more information on UUID numbers, see <http://www.opengroup.org/onlinepubs/9629399/apdx.htm>.

CHAPTER 3

Example

- [Examples: Configuring Application Protocols on page 55](#)

Examples: Configuring Application Protocols

The following example shows an application protocol definition describing a special FTP application running on port 78:

```
[edit applications]
application my-ftp-app {
  application-protocol ftp;
  protocol tcp;
  destination-port 78;
  timeout 100; # inactivity timeout for FTP service
}
```

The following example shows a special ICMP protocol (**application-protocol icmp**) of type 8 (ICMP echo):

```
[edit applications]
application icmp-app {
  application-protocol icmp;
  protocol icmp;
  icmp-type icmp-echo;
}
```

The following example shows a possible application set:

```
[edit applications]
application-set basic {
  http;
  ftp;
  telnet;
  nfs;
  icmp;
}
```

The software includes a predefined set of well-known application protocols. The set includes applications for which the TCP and UDP destination ports are already recognized by stateless firewall filters.

Related Documentation

- [ALG Descriptions on page 3](#)

- [Configuring Application Sets on page 37](#)
- [Configuring Application Protocol Properties on page 37](#)
- [Verifying the Output of ALG Sessions on page 57](#)

CHAPTER 4

Verification

- [Verifying the Output of ALG Sessions on page 57](#)

Verifying the Output of ALG Sessions

This section contains examples of successful output from ALG sessions and information on system log configuration. You can compare the results of your sessions to check whether the configurations are functioning correctly.

- [FTP Example on page 57](#)
- [RTSP ALG Example on page 60](#)
- [System Log Messages on page 62](#)

FTP Example

This example analyzes the output during an active FTP session. It consists of four different flows; two are control flows and two are data flows. The example consists of the following parts:

- [Sample Output on page 57](#)
- [FTP System Log Messages on page 58](#)
- [Analysis on page 59](#)
- [Troubleshooting Questions on page 59](#)

Sample Output

The following is a complete sample output from the **show services stateful-firewall conversations application-protocol ftp** operational mode command:

```
user@host>show services stateful-firewall conversations application-protocol ftp
Interface: ms-1/3/0, Service set: CLBJI1-AAF001
Conversation: ALG protocol: ftp
  Number of initiators: 2, Number of responders: 2
Flow      State  Dir      Frm count
TCP       1.1.79.2:14083 -> 2.2.2.2:21 Watch I      13
  NAT source 1.1.79.2:14083 -> 194.250.1.237:50118
TCP       1.1.79.2:14104 -> 2.2.2.2:20 Forward I      3
  NAT source 1.1.79.2:14104 -> 194.250.1.237:50119
TCP       2.2.2.2:21 -> 194.250.1.237:50118 Watch 0      12
  NAT dest 194.250.1.237:50118 -> 1.1.79.2:14083
```

```
TCP          2.2.2.2:20    -> 194.250.1.237:50119 Forward  0      5
NAT dest     194.250.1.237:50119 -> 1.1.79.2:14104
```

For each flow, the first line shows flow information, including protocol (TCP), source address, source port, destination address, destination port, flow state, direction, and frame count.

- The state of a flow can be **Watch**, **Forward**, or **Drop**:
 - A **Watch** flow state indicates that the control flow is monitored by the ALG for information in the payload. NAT processing is performed on the header and payload as needed.
 - A **Forward** flow forwards the packets without monitoring the payload. NAT is performed on the header as needed.
 - A **Drop** flow drops any packet that matches the 5 tuple.
- The frame count (**Fr m count**) shows the number of packets that were processed on that flow.

The second line shows the NAT information.

- **source** indicates source NAT.
- **dest** indicates destination NAT.
- The first address and port in the NAT line are the original address and port being translated for that flow.
- The second address and port in the NAT line are the translated address and port for that flow.

FTP System Log Messages

System log messages are generated during an FTP session. For more information about system logs, see [“System Log Messages” on page 62](#).

The following system log messages are generated during creation of the FTP control flow:

- Rule Accept system log:

```
Oct 27 11:42:54 (FPC Slot 1, PIC Slot 1) {ss_ftp}[FWNAT]: ASP_SFW_RULE_ACCEPT:
proto 6 (TCP) application: ftp, fe-3/3/3.0:1.1.1.2:4450 -> 2.2.2.2:21, Match SFW accept
rule-set:, rule: ftp, term: 1
```
- Create Accept Flow system log:

```
Oct 27 11:42:54 (FPC Slot 1, PIC Slot 1) {ss_ftp}[FWNAT]:
ASP_SFW_CREATE_ACCEPT_FLOW: proto 6 (TCP) application: ftp,
fe-3/3/3.0:1.1.1.2:4450 -> 2.2.2.2:21, creating forward or watch flow
```
- System log for data flow creation:

```
Oct 27 11:43:30 (FPC Slot 1, PIC Slot 1) {ss_ftp}[FWNAT]:
ASP_SFW_FTP_ACTIVE_ACCEPT: proto 6 (TCP) application: ftp, so-2/1/2.0:2.2.2.2:20
-> 1.1.1.2:50726, Creating FTP active mode forward flow
```


Analysis

Control Flows

The control flows are established after the three-way handshake is complete.

- Control flow from FTP client to FTP server. TCP destination port is 21.

```
TCP          1.1.79.2:14083 ->      2.2.2.2:21    Watch    I
13
NAT source   1.1.79.2:14083  ->    194.250.1.237:50118
```

- Control flow from FTP server to FTP client. TCP source port is 21.

```
TCP          2.2.2.2:21    ->    194.250.1.237:50118 Watch    O
12
NAT dest     194.250.1.237:50118 ->      1.1.79.2:14083
```

Data Flows

A data port of 20 is negotiated for data transfer during the course of the FTP control protocol. These two flows are data flows between the FTP client and the FTP server:

```
TCP          1.1.79.2:14104 ->      2.2.2.2:20    Forward  I           3
NAT source   1.1.79.2:14104  ->    194.250.1.237:50119
TCP          2.2.2.2:20    ->    194.250.1.237:50119 Forward  O           5
NAT dest     194.250.1.237:50119 ->      1.1.79.2:14104
```

Troubleshooting Questions

- How do I know if the FTP ALG is active?
 - The ALG protocol field in the conversation should display **ftp**.
 - There should be a valid frame count (**Frm count**) in the control flows.
 - A valid frame count in the data flows indicates that data transfer has taken place.
- What do I need to check if the FTP connection is established but data transfer does not take place?
 - Most probably, the control connection is up, but the data connection is down.
 - Check the conversations output to determine whether both the control and data flows are present.
- How do I interpret each flow? What does each flow mean?
 - FTP control flow initiator flow—Flow with destination port 21
 - FTP control flow responder flow—Flow with source port ;21
 - FTP data flow initiator flow—Flow with destination port 20
 - FTP data flow responder flow—Flow with source port 20

RTSP ALG Example

The following is an example of an RTSP conversation. The application uses the RTSP protocol for control connection. Once the connection is set up, the media is sent using UDP protocol (RTP).

This example consists of the following:

- [Sample Output on page 60](#)
- [Analysis on page 60](#)
- [Troubleshooting Questions on page 60](#)

Sample Output

Here is the output from the **show services stateful-firewall conversations** operational mode command:

```
user@host# show services stateful-firewall conversations
Interface: ms-3/2/0, Service set: svc_set
Conversation: ALG protocol: rtsp
  Number of initiators: 5, Number of responders: 5
```

Flow			State	Dir	Frm count	
TCP	1.1.1.3:58795	->	2.2.2.2:554	Watch	I	7
UDP	1.1.1.3:1028	->	2.2.2.2:1028	Forward	I	0
UDP	1.1.1.3:1029	->	2.2.2.2:1029	Forward	I	0
UDP	1.1.1.3:1030	->	2.2.2.2:1030	Forward	I	0
UDP	1.1.1.3:1031	->	2.2.2.2:1031	Forward	I	0
TCP	2.2.2.2:554	->	1.1.1.3:58795	Watch	O	5
UDP	2.2.2.2:1028	->	1.1.1.3:1028	Forward	O	6
UDP	2.2.2.2:1029	->	1.1.1.3:1029	Forward	O	0
UDP	2.2.2.2:1030	->	1.1.1.3:1030	Forward	O	3
UDP	2.2.2.2:1031	->	1.1.1.3:1031	Forward	O	0

Analysis

An RTSP conversation should consist of TCP flows corresponding to the RTSP control connection. There should be two flows, one in each direction, from client to server and from server to client:

TCP	1.1.1.3:58795	->	2.2.2.2:554	Watch	I	7
TCP	2.2.2.2:554	->	1.1.1.3:58795	Watch	O	5

- The RTSP control connection for the initiator flow is sent from destination port 554.
- The RTSP control connection for the responder flow is sent from source port 554.

The UDP flows correspond to RTP media sent over the RTSP connection.

Troubleshooting Questions

1. Media does not work when the RTSP ALG is configured. What do I do?
 - Check RTSP conversations to see whether both TCP and UDP flows exist.
 - The ALG protocol should be displayed as **rtsp**.



NOTE: The state of the flow is displayed as **Watch**, because the ALG processing is taking place and the client is essentially “watching” or processing payload corresponding to the application. For FTP and RTSP ALG flows, the control connections are always Watch flows.

2. How do I check for ALG errors?

- You can check for errors by issuing the following command. Each ALG has a separate field for ALG packet errors.

```
user@host# show services stateful-firewall statistics extensive
Interface: ms-3/2/0
Service set: svc_set
New flows:
  Accepts: 1347, Discards: 0, Rejects: 0
Existing flows:
  Accepts: 144187, Discards: 0, Rejects: 0
Drops:
  IP option: 0, TCP SYN defense: 0
  NAT ports exhausted: 0
Errors:
  IP: 0, TCP: 276
  UDP: 0, ICMP: 0
  Non-IP packets: 0, ALG: 0
IP errors:
  IP packet length inconsistencies: 0
  Minimum IP header length check failures: 0
  Reassembled packet exceeds maximum IP length: 0
  Illegal source address: 0
  Illegal destination address: 0
  TTL zero errors: 0, Illegal IP protocol number (0 or 255): 0
  Land attack: 0
  Non-IPv4 packets: 0, Bad checksum: 0
  Illegal IP fragment length: 0
  IP fragment overlap: 0
  IP fragment reassembly timeout: 0
  Unknown: 0
TCP errors:
  TCP header length inconsistencies: 0
  Source or destination port number is zero: 0
  Illegal sequence number and flags combinations: 0
  SYN attack (multiple SYN messages seen for the same flow): 276
  First packet not a SYN message: 0
  TCP port scan (TCP handshake, RST seen from server for SYN): 0
  Bad SYN cookie response: 0
UDP errors:
  IP data length less than minimum UDP header length (8 bytes): 0
  Source or destination port number is zero: 0
  UDP port scan (ICMP error seen for UDP flow): 0
ICMP errors:
  IP data length less than minimum ICMP header length (8 bytes): 0
  ICMP error length inconsistencies: 0
  Duplicate ping sequence number: 0
  Mismatched ping sequence number: 0
ALG errors:
  BOOTP: 0, DCE-RPC: 0, DCE-RPC portmap: 0
  DNS: 0, Exec: 0, FTP: 0
  ICMP: 0
```

```
Login: 0, NetBIOS: 0, NetShow: 0
RPC: 0, RPC portmap: 0
RTSP: 0, Shell: 0
SNMP: 0, SQLNet: 0, TFTP: 0
Traceroute: 0
```

System Log Messages

Enabling system log generation and checking the system log are also helpful for ALG flow analysis. This section contains the following:

- [System Log Configuration on page 62](#)
- [System Log Output on page 63](#)

System Log Configuration

You can configure the enabling of system log messages at a number of different levels in the Junos OS CLI. As shown in the following sample configurations, the choice of level depends on how specific you want the event logging to be and what options you want to include. For details on the configuration options, see the *Junos OS Administration Library for Routing Devices* (system level) or the *Junos OS Services Interfaces Library for Routing Devices* (all other levels).

1. At the topmost global level:

```
user@host# show system syslog
file messages {
    any any;
}
```

2. At the service set level:

```
user@host# show services service-set svc_set
syslog {
    host local {
        services any;
    }
}
stateful-firewall-rules allow_rtsp;
interface-service {
    service-interface ms-3/2/0;
}
```

3. At the service rule level:

```
user@host# show services stateful-firewall rule allow_rtsp
match-direction input-output;
term 0 {
    from {
        applications junos-rtsp;
    }
    then {
        accept;
        syslog;
    }
}
```

System Log Output

System log messages are generated during flow creation, as shown in the following examples:

The following system log message indicates that the ASP matched an accept rule:

```
Oct 25 16:11:37 (FPC Slot 3, PIC Slot 2) {svc_set}[FWNAT]: ASP_SFW_RULE_ACCEPT:
  proto 6 (TCP) application: rtsp, ge-2/0/1.0:1.1.1.2:35595 -> 2.2.2.2:554, Match SFW accept
  rule-set: , rule: allow_rtsp, term: 0
```

For a complete listing of system log messages, see the *Junos OS System Log Messages Reference*.

Related Documentation

- [ALG Descriptions on page 3](#)
- [Configuring Application Sets on page 37](#)
- [Configuring Application Protocol Properties on page 37](#)
- [Examples: Configuring Application Protocols on page 55](#)

CHAPTER 5

Configuration Statements

application

Syntax `application application-name {
 application-protocol protocol-name;
 destination-port port-number;
 icmp-code value;
 icmp-type value;
 inactivity-timeout value;
 protocol type;
 rpc-program-number number;
 snmp-command command;
 source-port port-number;
 ttl-threshold number;
 uuid hex-value;
 }`

Hierarchy Level `[edit applications],`
 `[edit applications application-set application-set-name]`

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure properties of an application and whether to include it in an application set.

Options *application-name*—Identifier of the application.

 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

- [ALG Descriptions on page 3](#)
- [Configuring Application Sets on page 37](#)
- [Configuring Application Protocol Properties on page 37](#)
- [Examples: Configuring Application Protocols on page 55](#)
- [Verifying the Output of ALG Sessions on page 57](#)

application-protocol

Syntax	<code>application-protocol <i>protocol-name</i>;</code>
Hierarchy Level	[edit applications application <i>application-name</i>]
Release Information	Statement introduced before Junos OS Release 7.4. login options introduced in Junos OS Release 7.4. ip option introduced in Junos OS Release 8.2.
Description	Identify the application protocol name. Application protocols are also called application layer gateways (ALGs).
Options	<p><i>protocol-name</i>—Name of the protocol. The following protocols are supported:</p> <ul style="list-style-type: none">bootp—Bootstrap protocoldce-rpc—DCE RPCdce-rpc-portmap—DCE RPC portmapdns—Domain Name Serviceexec—Remote Execution Protocolftp—File Transfer Protocolh323—H.323icmp—ICMPiiop—Internet Inter-ORB Protocolip—IPlogin—Loginnetbios—NetBIOSnetshow—NetShowpptp—Point-to-Point Tunneling Protocolrealaudio—RealAudiorpc—RPCrpc-portmap—RPC portmaprtsp—Real Time Streaming Protocolshell—Shellsip—Session Initiation Protocolsnmp—SNMPsqlnet—SQLNettalk—Talk Program

tftp—Trivial File Transfer Protocol

traceroute—Traceroute

winframe—WinFrame

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- [ALG Descriptions on page 3](#)
- [Configuring Application Sets on page 37](#)
- [Configuring Application Protocol Properties on page 37](#)
- [Examples: Configuring Application Protocols on page 55](#)
- [Verifying the Output of ALG Sessions on page 57](#)

application-set

Syntax `application-set application-set-name {
 application application-name;
}`

Hierarchy Level [edit [applications](#)]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure one or more applications to include in an application set.

Options *application-set-name*—Identifier of an application set.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- [ALG Descriptions on page 3](#)
- [Configuring Application Sets on page 37](#)
- [Configuring Application Protocol Properties on page 37](#)
- [Examples: Configuring Application Protocols on page 55](#)
- [Verifying the Output of ALG Sessions on page 57](#)

applications

Syntax	<code>applications { ... }</code>
Hierarchy Level	[edit]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Define the applications used in services.
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">• ALG Descriptions on page 3• Configuring Application Sets on page 37• Configuring Application Protocol Properties on page 37• Examples: Configuring Application Protocols on page 55• Verifying the Output of ALG Sessions on page 57

destination-port

Syntax	<code>destination-port <i>port-value</i>;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) destination port number.
Options	<i>port-value</i> —Identifier for the port. For a complete list, see “Configuring Source and Destination Ports” on page 43 .
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">• ALG Descriptions on page 3• Configuring Application Sets on page 37• Configuring Application Protocol Properties on page 37• Examples: Configuring Application Protocols on page 55• Verifying the Output of ALG Sessions on page 57

icmp-code

Syntax	<code>icmp-code value;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Internet Control Message Protocol (ICMP) code value.
Options	value —The ICMP code value. For a complete list, see “ Configuring the ICMP Code and Type ” on page 41.
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"> • ALG Descriptions on page 3 • Configuring Application Sets on page 37 • Configuring the ICMP Code and Type on page 41 • Examples: Configuring Application Protocols on page 55 • Verifying the Output of ALG Sessions on page 57

icmp-type

Syntax	<code>icmp-type value;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	ICMP packet type value.
Options	value —The ICMP type value, such as echo or echo-reply . For a complete list, see “ Configuring the ICMP Code and Type ” on page 41.
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"> • ALG Descriptions on page 3 • Configuring Application Sets on page 37 • Configuring the ICMP Code and Type on page 41 • Examples: Configuring Application Protocols on page 55 • Verifying the Output of ALG Sessions on page 57

inactivity-timeout

Syntax	<code>inactivity-timeout <i>seconds</i>;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Inactivity timeout period, in seconds.
Options	<i>seconds</i> —Length of time the application is inactive before it times out. Default: 30 seconds
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">• ALG Descriptions on page 3• Configuring Application Sets on page 37• Configuring the Inactivity Timeout Period on page 46• Examples: Configuring Application Protocols on page 55• Verifying the Output of ALG Sessions on page 57

learn-sip-register

Syntax	<code>learn-sip-register;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced in Junos OS Release 7.4.
Description	Activate SIP register to accept potential incoming SIP calls.
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">• ALG Descriptions on page 3• Configuring Application Sets on page 37• Configuring SIP on page 37• Examples: Configuring Application Protocols on page 55• Verifying the Output of ALG Sessions on page 57

protocol

Syntax	<code>protocol type;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Networking protocol type or number.
Options	<p>type—Networking protocol type. The following text values are supported:</p> <ul style="list-style-type: none"> ah egp esp gre icmp icmp6 igmp ipip ospf pim rsvp tcp udp



NOTE: IP version 6 (IPv6) is not supported as a network protocol in application definitions.

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"> • ALG Descriptions on page 3 • Configuring Application Sets on page 37 • Configuring Application Protocol Properties on page 37 • Examples: Configuring Application Protocols on page 55 • Verifying the Output of ALG Sessions on page 57

rpc-program-number

Syntax	<code>rpc-program-number <i>number</i>;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Remote procedure call (RPC) or Distributed Computing Environment (DCE) value.
Options	<i>number</i> —RPC or DCE program value. Range: 100,000 through 400,000
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">• ALG Descriptions on page 3• Configuring an RPC Program Number on page 53• Examples: Configuring Application Protocols on page 55• Verifying the Output of ALG Sessions on page 57

sip-call-hold-timeout

Syntax	<code>sip-call-hold-timeout <i>seconds</i>;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced in Junos OS Release 7.4.
Description	Timeout period for SIP calls placed on hold, in seconds.
Options	<i>seconds</i> —Length of time the application holds a SIP call open before it times out. Default: 7200 seconds Range: 0 through 36,000 seconds (10 hours)
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">• ALG Descriptions on page 3• Configuring SIP on page 37• Examples: Configuring Application Protocols on page 55• Verifying the Output of ALG Sessions on page 57

snmp-command

Syntax	<code>snmp-command <i>command</i>;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	SNMP command format.
Options	<i>command</i> —Supported commands are SNMP get , get-next , set , and trap .
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"> • ALG Descriptions on page 3 • Configuring an SNMP Command for Packet Matching on page 53 • Examples: Configuring Application Protocols on page 55 • Verifying the Output of ALG Sessions on page 57

source-port

Syntax	<code>source-port <i>port-number</i>;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Source port identifier.
Options	<i>port-value</i> —Identifier for the port. For a complete list, see “ Configuring Source and Destination Ports ” on page 43.
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"> • ALG Descriptions on page 3 • Configuring Application Protocol Properties on page 37 • Configuring Source and Destination Ports on page 43 • Verifying the Output of ALG Sessions on page 57

ttl-threshold

Syntax	<code>ttl-threshold <i>number</i>;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Specify the traceroute time-to-live (TTL) threshold value. This value sets the acceptable level of network penetration for trace routing.
Options	<i>number</i> —TTL threshold value.
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">• ALG Descriptions on page 3• Configuring the TTL Threshold on page 54.• Examples: Configuring Application Protocols on page 55• Verifying the Output of ALG Sessions on page 57

uuid

Syntax	<code>uuid <i>hex-value</i>;</code>
Hierarchy Level	[edit applications application application-name]
Release Information	Statement introduced before Junos OS Release 7.4.
Description	Specify the Universal Unique Identifier (UUID) for DCE RPC objects.
Options	<i>hex-value</i> —Hexadecimal value.
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">• ALG Descriptions on page 3• Configuring a Universal Unique Identifier on page 54• Examples: Configuring Application Protocols on page 55• Verifying the Output of ALG Sessions on page 57

PART 3

Index

- [Index on page 77](#)

Index

Symbols

#, comments in configuration statements.....	xii
(), in syntax descriptions.....	xii
< >, in syntax descriptions.....	xii
[], in configuration statements.....	xii
{ }, in configuration statements.....	xii
(pipe), in syntax descriptions.....	xii

A

ALGs	
configuring.....	38
ALGsJ	
default.....	30
application statement.....	65
usage guidelines.....	37
application-protocol statement.....	66
usage guidelines.....	38
application-set statement.....	67
usage guidelines.....	37
applications	
example configuration.....	55
applications statement	
applications hierarchy.....	68

B

braces, in configuration statements.....	xii
brackets	
angle, in syntax descriptions.....	xii
square, in configuration statements.....	xii

C

CGNAT	
ALGs.....	30
comments, in configuration statements.....	xii
conventions	
text and syntax.....	xi
curly braces, in configuration statements.....	xii
customer support.....	xiii
contacting JTAC.....	xiii

D

destination-port statement	
applications.....	68
RPM.....	68
usage guidelines.....	43
documentation	
comments on.....	xiii

F

font conventions.....	xi
-----------------------	----

I

icmp-code statement.....	69
usage guidelines.....	41
icmp-type statement.....	69
usage guidelines.....	41
inactivity-timeout statement.....	70
usage guidelines.....	46

L

learn-sip-register statement.....	70
-----------------------------------	----

M

manuals	
comments on.....	xiii

N

NAT	
ALGs.....	30

P

parentheses, in syntax descriptions.....	xii
protocol statement	
applications.....	71
usage guidelines.....	40

R

rpc-program-number statement.....	72
usage guidelines.....	53

S

sip-call-hold-timeout statement.....	72
snmp-command statement.....	73
usage guidelines.....	53
source-port statement	
RPM.....	73
usage guidelines.....	43
support, technical See technical support	
syntax conventions.....	xi

T

technical support	
contacting JTAC.....	xiii
time-to-live threshold.....	54
ttl-threshold statement.....	74
usage guidelines.....	54

U

Universal Unique Identifier.....	54
uuid statement.....	74
usage guidelines.....	54