Application Properties

Published: 2013-02-15

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

Application Properties

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

List of Tables

About the Documentation

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.xsl;
       }
     }
   }
   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.xsl; }
   ```

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
Italic text like this	• Introduces or emphasizes important new terms. • Identifies book names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS System Basics Configuration Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	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** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Enclose optional keywords or variables.	**stub <default-metric** ***metric*****>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	**community name members [** ***community-ids*** **]**

Indention and braces ({ })	Identify a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
J-Web GUI Conventions		
Bold text like this	Represents J-Web graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of J-Web selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

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

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

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need 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

CHAPTER 1

Application-Level Gateway

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

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
IIOP	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

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 throught 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.

IIOP

The Oracle Application Server NameServer Internet Inter-ORB Protocol (IIOP). This ALG is used in Common Object Request Broker Architecture (CORBA) based on distributed computing. Even though CORBA and IIOP are Object Management Group (OMG) standards, there is no fixed port assigned for IIOP. 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 IIOP be configured for TCP port 1975 for Java VM IIOP, and 3075 for CORBA applications ORBIX, a CORBA framework from Iona Technologies.

IP

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.
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.
rpc-ypbind	NIS binding process. For details, see the UNIX man page for **ypbind**.	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-ypbind** 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.

Table 5: Supported RPC Services *(continued)*

Name	Description	Comments
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.

Starting with Junos OS Release 11.4, the SIP ALG supports Network Address Translation (NAT) and stateful firewall configuration on JSF. The SIP ALG supports the following features:

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

NOTE: The SIP ALG does not support destination NAT, class of service (CoS), multicast, or IP version 6 (IPv6).

At present, the SIP ALG does not support the following features:

- Encryption and authentication of SIP messages
- Transport of SIP messages over TCP

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*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-cvspserver {
    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;
    }
}
```

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 23
- RTSP ALG Example on page 26
- System Log Messages on page 28

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 24
- FTP System Log Messages on page 24
- Analysis on page 25
- Troubleshooting Questions on page 25

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    O              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  O               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 (**Frm 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 28.

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

1. 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.
2. 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.

3. 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 26
- Analysis on page 26
- Troubleshooting Questions on page 27

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

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 System Basics Configuration Guide (system level) or the Junos Services Interfaces Configuration Release 12.3 (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*.

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 39; 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;
        }
      }
    }
  }
}
```

PART 2

Configuration

CHAPTER 2

Configuration Tasks

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 51.

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 39.

This section includes the following tasks for configuring applications:

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 6 on page 40 shows the list of supported protocols. For more information about specific protocols, see "ALG Descriptions" on page 3.

Table 6: 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.
FTP	**ftp**	Requires the **protocol** statement to have the value **tcp** or to be unspecified. Requires a **destination-port** value.
H.323	**h323**	–

Table 6: Application Protocols Supported by Services Interfaces *(continued)*

Protocol Name	CLI Value	Comments
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**	
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.

Table 6: Application Protocols Supported by Services Interfaces *(continued)*

Protocol Name	CLI Value	Comments
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 Network Address Translation.

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 7 on page 42 shows the list of the supported protocols.

Table 7: 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**.
Internet Group Management Protocol (IGMP)	**igmp**	–
IP in IP	**ipip**	–

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

Network Protocol Type	CLI Value	Comments
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**.
Virtual Router Redundancy Protocol (VRRP)	**vrrp**	–

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 Network Address Translation.

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 8 on page 44 shows the list of supported ICMP values.

Table 8: ICMP Codes and Types Supported by Services Interfaces

CLI Statement	Description
icmp-code	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 Routing Policy Configuration Guide. 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: parameter-problem: **ip-header-bad** (0), **required-option-missing** (1) redirect: **redirect-for-host** (1), **redirect-for-network** (0), **redirect-for-tos-and-host** (3), **redirect-for-tos-and-net** (2) time-exceeded: **ttl-eq-zero-during-reassembly** (1), **ttl-eq-zero-during-transit** (0) unreachable: **communication-prohibited-by-filtering** (13), **destination-host-prohibited** (10), **destination-host-unknown** (7), **destination-network-prohibited** (9), **destination-network-unknown** (6), **fragmentation-needed** (4), **host-precedence-violation** (14), **host-unreachable** (1), **host-unreachable-for-TOS** (12), **network-unreachable** (0), **network-unreachable-for-TOS** (11), **port-unreachable** (3), **precedence-cutoff-in-effect** (15), **protocol-unreachable** (2), **source-host-isolated** (8), **source-route-failed** (5)
icmp-type	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 Routing Policy Configuration Guide. 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).

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 6 on page 40.

You can specify either a numeric value or one of the text synonyms listed in Table 9 on page 45.

Table 9: 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 9: 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
mobilip-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 9: 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
xdmcp	177
zephyr-clt	2103
zephyr-hm	2104

For more information about matching criteria, see the Routing Policy Configuration Guide.

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 supported standard is described in RFC 3261, SIP: Session Initiation Protocol, which includes stateful firewall and Network Address Translation (NAT) support for SIP dialogs and UDP IPv4 transport of SIP messages.

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 40. 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).

Limitations

The following limitations apply to configuration of the SIP ALG:

- Only the methods described in RFC 3261 are supported.
- Only SIP version 3 is supported.
- TCP is not supported as a transport mechanism for signaling messages.
- 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.
- When 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. In such cases, you should not configure the ALG.

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 40.

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 40.

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 40.

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 40. For more information on UUID numbers, see **http://www.opengroup.org/onlinepubs/9629399/apdxa.htm**.

CHAPTER 3

Example

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.

CHAPTER 4

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.

Usage Guidelines

See "Configuring Application Protocol Properties" on page 39.

Required Privilege Level

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

application-protocol

Syntax application-protocol *protocol-name*;

Hierarchy Level [edit applications application *application-name*]

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 *protocol-name*—Name of the protocol. The following protocols are supported:

bootp—Bootstrap protocol

dce-rpc—DCE RPC

dce-rpc-portmap—DCE RPC portmap

dns—Domain Name Service

exec—Remote Execution Protocol

ftp—File Transfer Protocol

h323—H.323

icmp—ICMP

iiop—Internet Inter-ORB Protocol

ip—IP

login—Login

netbios—NetBIOS

netshow—NetShow

pptp—Point-to-Point Tunneling Protocol

realaudio—RealAudio

rpc—RPC

rpc-portmap—RPC portmap

rtsp—Real Time Streaming Protocol

shell—Shell

sip—Session Initiation Protocol

snmp—SNMP

sqlnet—SQLNet

talk–Talk Program

tftp—Trivial File Transfer Protocol

traceroute—Traceroute

winframe—WinFrame

Usage Guidelines See "Configuring an Application Protocol" on page 40.

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

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.

Usage Guidelines See "Configuring Application Sets" on page 39.

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

applications

Syntax applications { ... }

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

- Application Properties

destination-port

Syntax destination-port *port-value*;

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 ***port-value***—Identifier for the port. For a complete list, see "Configuring Source and Destination Ports" on page 45.

Usage Guidelines See "Configuring Source and Destination Ports" on page 45.

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

icmp-code

Syntax icmp-code *value*;

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 43.

Usage Guidelines See "Configuring the ICMP Code and Type" on page 43.

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

icmp-type

Syntax icmp-type *value*;

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 43.

Usage Guidelines See "Configuring the ICMP Code and Type" on page 43.

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

inactivity-timeout

Syntax inactivity-timeout *seconds*;

Hierarchy Level [edit applications application *application-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Inactivity timeout period, in seconds.

Options ***seconds***—Length of time the application is inactive before it times out.
Default: 30 seconds

Usage Guidelines See "Configuring the Inactivity Timeout Period" on page 48.

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

learn-sip-register

Syntax learn-sip-register;

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.

Usage Guidelines See "Configuring SIP" on page 39.

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

protocol

Syntax protocol *type*;

Hierarchy Level [edit applications application *application-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Networking protocol type or number.

Options *type*—Networking protocol type. The following text values are supported:

- **ah**
- **egp**
- **esp**
- **gre**
- **icmp**
- **igmp**
- **ipip**
- **ospf**
- **pim**
- **rsvp**
- **tcp**
- **udp**
- **vrrp**

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

Usage Guidelines See "Configuring the Network Protocol" on page 42.

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

rpc-program-number

Syntax rpc-program-number *number*;

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 *number*—RPC or DCE program value.
Range: 100,000 through 400,000

Usage Guidelines See "Configuring an RPC Program Number" on page 50.

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

sip-call-hold-timeout

Syntax sip-call-hold-timeout *seconds*;

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 *seconds*—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)

Usage Guidelines See "Configuring SIP" on page 39.

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

snmp-command

Syntax snmp-command *command*;

Hierarchy Level [edit applications application *application-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description SNMP command format.

Options ***command***—Supported commands are SNMP **get**, **get-next**, **set**, and **trap**.

Usage Guidelines See "Configuring an SNMP Command for Packet Matching" on page 49.

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

source-port

Syntax source-port *port-number*;

Hierarchy Level [edit applications application *application-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Source port identifier.

Options ***port-value***—Identifier for the port. For a complete list, see "Configuring Source and Destination Ports" on page 45.

Usage Guidelines See "Configuring Source and Destination Ports" on page 45.

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

ttl-threshold

Syntax ttl-threshold *number*;

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 ***number***—TTL threshold value.

Usage Guidelines See "Configuring the TTL Threshold" on page 50.

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

uuid

Syntax uuid *hex-value*;

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 ***hex-value***—Hexadecimal value.

Usage Guidelines See "Configuring a Universal Unique Identifier" on page 50.

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

PART 3

Index

Index