

Application Layer Gateways User Guide

Published
2025-12-08

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

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

Application Layer Gateways User Guide
Copyright © 2025 Juniper Networks, Inc. All rights reserved.

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

[About This Guide | xiii](#)

1

Overview

[ALG Overview | 2](#)

[ALG Overview | 2](#)

[Understanding Custom ALG Services | 3](#)

[Understanding the IPv6 DNS ALG for Routing, NAT, and NAT-PT | 5](#)

[Understanding IPv6 Support in FTP ALG | 7](#)

[Understanding TAP Mode Support for ALG | 9](#)

[Enabling and Disabling ALG in TAP Mode | 9](#)

2

Data ALGs

[Understanding Data ALG Types | 13](#)

[DNS ALG | 15](#)

[DNS ALG Overview | 16](#)

[Example: Configuring the DNS ALG | 17](#)

[Requirements | 17](#)

[Overview | 17](#)

[Configuration | 18](#)

[Verification | 21](#)

[Understanding DNS and DDNS Doctoring | 23](#)

[Platform-Specific DNS ALG Behavior | 27](#)

[FTP ALG | 28](#)

[FTP ALG Overview | 28](#)

[Understanding FTP Commands | 30](#)

[Example: Configuring the FTP ALG | 32](#)

[Requirements | 32](#)

Overview	32
Configuration	32
Verification	36

IKE and ESP ALG | 37

Understanding the IKE and ESP ALG | 38

Example: Configuring the IKE and ESP ALG | 40

Requirements	40
Overview	40
Configuration	41
Verification	47

Example: Enabling the IKE and ESP ALG and Setting Timeouts | 49

Requirements	49
Overview	49
Configuration	50
Verification	51

Platform-Specific IKE ALG Behavior | 52

PPTP ALG | 53

Understanding the PPTP ALG | 53

Example: Configuring the PPTP ALG | 54

Requirements	55
Overview	55
Configuration	57
Verification	68

RPC ALG | 72

Understanding RPC ALGs | 72

Understanding Sun RPC ALGs | 73

Enabling Sun RPC ALGs | 74

Customizing Sun RPC Applications (CLI Procedure) | 74

Understanding Sun RPC Services | 75

Understanding Microsoft RPC ALGs | 78

Enabling Microsoft RPC ALGs | 79

Configuring the Microsoft RPC ALG | 80

Configuring the MS-RPC ALG with a Predefined Microsoft Application | 80

Configuring the MS-RPC ALG with a Wildcard UUID | 81

Configuring the MS-RPC ALG with a Specific UUID | 81

Understanding Microsoft RPC Services | 83

Customizing Microsoft RPC Applications (CLI Procedure) | 85

RSH ALG | 87

Understanding the RSH ALG | 87

Example: Configuring the RSH ALG | 88

Requirements | 88

Overview | 88

Configuration | 91

Verification | 103

RTSP ALG | 106

Understanding the RTSP ALG | 106

Understanding RTSP ALG Messages | 108

Understanding RTSP ALG Conversation and NAT | 110

Example: Configuring the RTSP ALG | 113

Requirements | 113

Overview | 113

Configuration | 114

Verification | 119

SQLNET ALG | 124

Understanding the SQLNET ALG | 124

Example: Configuring the SQLNET ALG | 125

Requirements | 125

Overview | 126

Configuration | 128

Verification | 140

TALK ALG | 144

Understanding the TALK ALG | 144

Example: Configuring the TALK ALG | 144

Requirements | 145

Overview | 145

Configuration | 148

Verification | 159

TFTP ALG | 163

Understanding the TFTP ALG | 164

Example: Configuring the TFTP ALG | 166

Requirements | 167

Overview | 167

Configuration | 167

Verification | 170

Platform-Specific TFTP ALG Behavior | 171

TWAMP ALG | 172

Understanding the Two-Way Active Measurement Protocol (TWAMP) Application Layer Gateway (ALG) | 173

Enabling the Two-Way Active Measurement Protocol (TWAMP) Application Layer Gateway (ALG) | 175

Understanding IPv6 ALG Support for ICMP | 176

Understanding 464XLAT ALG Traffic Support | 178

Understanding ALG Support for VRF Routing Instance | 186

VoIP ALGs

Understanding VoIP ALG Types | 188

VoIP DSCP Rewrite Rules | 189

Understanding VoIP DSCP Rewrite Rules | 189

Example: Configuring VoIP DSCP Rewrite Rules | 190

Requirements | 190

Overview | 190

Configuration | 191

Verification | 191

H.323 ALG | 192

Understanding H.323 ALG | 192

Understanding the Avaya H.323 ALG | 195

Example: Passing H.323 ALG Traffic to a Gatekeeper in the Private Zone | 197

Requirements | 197

Overview | 198

Configuration | 198

Verification | 202

Example: Passing H.323 ALG Traffic to a Gatekeeper in the External Zone | 204

Requirements | 204

Overview | 204

Configuration | 205

Verification | 209

Example: Using NAT with the H.323 ALG to Enable Incoming Calls | 211

Requirements | 211

Overview | 212

Configuration | 213

Verification | 219

Example: Using NAT with the H.323 ALG to Enable Outgoing Calls | 222

Requirements | 223

Overview | 223

Configuration | 224

Verification | 230

Example: Setting H.323 ALG Endpoint Registration Timeouts | 232

Requirements | 232

Overview | 233

Configuration | 233

Verification | 234

Example: Setting H.323 ALG Media Source Port Ranges | 234

Requirements | 234

Overview | 234

Configuration | 235

Verification | 236

Example: Configuring H.323 ALG DoS Attack Protection | 236

Requirements | 236

Overview | 236

Configuration | 236

Verification | 237

Understanding H.323 ALG Known Message Types | 238

Understanding H.323 ALG Unknown Message Types | 243

Example: Allowing Unknown H.323 ALG Message Types | 244

Requirements | 244

Overview | 245

Configuration | 245

Verification | 246

MGCP ALG | 247

Understanding the MGCP ALG | 247

MGCP ALG Configuration Overview | 254

Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs | 255

Requirements | 255

Overview | 255

Configuration | 257

Verification | 261

Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALG and NAT | 265

Requirements | 265

Overview | 266

Configuration | 267

Verification | 279

Understanding MGCP ALG Call Duration and Timeouts | 283

Example: Setting MGCP ALG Call Duration | 284

Requirements | 284

Overview | 284

Configuration | 284

Verification | 285

Example: Setting MGCP ALG Inactive Media Timeout | 286

Requirements | 286

Overview | 286

Configuration | 286

Verification | 287

Example: Setting MGCP ALG Transaction Timeout | 288

Requirements | 288

Overview | 288

Configuration | 288

Verification | 289

Example: Configuring MGCP ALG DoS Attack Protection | 290

Requirements | 290

Overview | 290

Configuration | 290

Verification | 291

Example: Allowing Unknown MGCP ALG Message Types | 292

Requirements | 292

Overview | 292

Configuration | 293

Verification | 294

SCCP ALG | 294

Understanding SCCP ALGs | 295

SCCP ALG Configuration Overview | 302

Example: Setting SCCP ALG Inactive Media Timeouts | 302

Requirements | 302

Overview | 302

Configuration | 303

Verification | 304

Example: Allowing Unknown SCCP ALG Message Types | 304

Requirements | 304

Overview | 304

Configuration | 305

Verification | 306

Example: Configuring SCCP ALG DoS Attack Protection | 306

Requirements | 306

Overview | 306

Configuration | 307

Verification | 308

Example: Configuring the SCCP ALG Call Manager or TFTP Server in the Private Zone | 308

Requirements | 308

Overview | 308

Configuration | 311

Verification | 317

Verifying SCCP ALG Configurations | 321

Verifying SCCP ALG | 321

Verifying SCCP ALG Calls | 322

Verifying SCCP ALG Call Details | 323

Verifying SCCP ALG Counters | 325

SIP ALG | 326

Understanding the SIP ALG | 327

Understanding SIP ALG Hold Resources | 337

Understanding the SIP ALG and NAT | 338

Example: Setting SIP ALG Call Duration and Timeouts | 352

Requirements | 352

Overview | 352

Configuration | 353

Verification | 354

Example: Configuring SIP ALG DoS Attack Protection | 354

Requirements | 354

Overview | 354

Configuration | 355

Verification | 356

Example: Allowing Unknown SIP ALG Message Types | 356

Requirements | 356

Overview | 357

Configuration | 357

Verification | 358

Example: Configuring Interface Source NAT for Incoming SIP Calls | 358

Requirements | 358

Overview | 358

Configuration | 360

Verification | 365

Example: Decreasing Network Complexity by Configuring a Source NAT Pool for Incoming SIP Calls | 367

Requirements | 367

Overview | 367

Configuration | 370

Verification | 375

Example: Configuring Static NAT for Incoming SIP Calls | 380

Requirements | 380

Overview | 380

Configuration | 383

Verification | 388

Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone | 391

Requirements | 391

Overview | 391

Configuration | 394

Verification | 399

Example: Configuring a Three-Zone SIP ALG and NAT Scenario | 403

Requirements | 403

Overview | 403

Configuration | 406

Verification | 414

Configuration Statements and Operational Commands

Junos CLI Reference Overview | 420

About This Guide

Use this guide to configure and monitor application layer gateway (ALG) to manage application protocols such as H.323, FTP, Session Initiation Protocol (SIP), and ICMP.

1

CHAPTER

Overview

IN THIS CHAPTER

- [ALG Overview | 2](#)
-

ALG Overview

IN THIS SECTION

- [ALG Overview | 2](#)
- [Understanding Custom ALG Services | 3](#)
- [Understanding the IPv6 DNS ALG for Routing, NAT, and NAT-PT | 5](#)
- [Understanding IPv6 Support in FTP ALG | 7](#)
- [Understanding TAP Mode Support for ALG | 9](#)
- [Enabling and Disabling ALG in TAP Mode | 9](#)

An Application Layer Gateway (ALG) enables the gateway to parse application layer payloads and take decisions whether to allow or deny traffic to the application server. ALGs supports the applications such as Transfer Protocol (FTP) and various IP protocols that use the application layer payload to communicate the dynamic Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) ports on which the applications open data connections.

ALG Overview

An Application Layer Gateway (ALG) is a software component that is designed to manage specific protocols such as Session Initiation Protocol (SIP) or FTP on Juniper Networks devices running Junos OS. The ALG module is responsible for Application-Layer aware packet processing on switches.

ALG functionality can be triggered either by a service or application configured in the security policy:

- A service is an object that identifies an application protocol using Layer 4 information (such as standard and accepted TCP and UDP port numbers) for an application service (such as Telnet, FTP, and SMTP).
- An application specifies the Layer 7 application that maps to a Layer 4 service.

A predefined service already has a mapping to a Layer 7 application. However, for custom services, you must link the service to an application explicitly, especially if you want the policy to apply an ALG.

Application Support Layer (ASL) is used by System Log Messages. Refer to [System Log Messages](#) for more information.

ALGs for packets destined to well-known ports are triggered by service type. The ALG intercepts and analyzes the specified traffic, allocates resources, and defines dynamic policies to permit the traffic to pass securely through the device:

1. When a packet arrives at the device, the flow module forwards the packet according to the security rule set in the policy.
2. If a policy is found to permit the packet, the associated service type or application type is assigned and a session is created for this type of traffic.
3. If a session is found for the packet, no policy rule match is needed. The ALG module is triggered if that particular service or application type requires the supported ALG processing.

The ALG also inspects the packet for embedded IP address and port information in the packet payload, and performs Network Address Translation (NAT) processing if necessary. A message buffer is allocated only when the packet is ready to process. The buffer is freed after the packet completes ALG handling, including modifying the payload, performing NAT, opening a pinhole for a new connection between a client and a server, and transferring data between a client and a server located on opposite sides of a Juniper Networks device

The maximum size of the jbuf is 9 Kb. If the message buffer size is more than 9 Kb, the entire message cannot be transferred to the ALG packet handler. This causes subsequent packets in the session to bypass ALG handling, resulting in a transaction failure. The ALG message buffer optimization is enhanced to reduce high memory consumption.

The ALG also opens a gate for the IP address and port number to permit data exchange for the control and data sessions. The control session and data session can be coupled to have the same timeout value, or they can be independent.

ALGs are supported on chassis clusters.

SEE ALSO

[Understanding H.323 ALG | 192](#)

[Understanding the SIP ALG | 327](#)

[Understanding SCCP ALGs | 295](#)

Understanding Custom ALG Services

By default, ALGs are bound to predefined services. For example, the FTP ALG is bound to `junos-ftp`, the RTSP ALG is bound to `junos-rtsp`, and so on.

A predefined service already has a mapping to a Layer 7 application. However, for custom services, you must link the service to an application explicitly, especially if you want the policy to apply an ALG.

When you apply predefined services to your policy, traffic matching the service will be sent to its corresponding ALG for further processing. However, under some circumstances, you might need to define custom services to achieve the following:

- Utilize the ALG handler to process special traffic, with customer-specified protocols, destination ports and so on.
- Permit traffic but bypass ALG processing, when traffic matches predefined services that bind with ALG.
- Add more applications to the current ALG's application set.

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

The three usages of custom services are illustrated below, considering MS-RPC ALG as an example:

- **Utilize the ALG handler to process special traffic:**

```
[edit]
user@host# set applications application customer-msrpc application-protocol ms-rpc
user@host# set applications application customer-msrpc protocol tcp
user@host# set applications application customer-msrpc destination-port 6000
```

Traffic with TCP destination port 6000 will be sent to MS-RPC ALG for further processing.

- **Permit traffic but bypass ALG processing:**

```
[edit]
user@host# set applications application customer-ignore application-protocol ignore
user@host# set applications application customer-ignore protocol tcp
user@host# set applications application customer-ignore destination-port 135
```

All ALGs will be ignored by traffic with TCP destination port 135.

- **Add more applications to an ALG's application set**—To add applications such as MS-RPC or Sun RPC services, which are not predefined on the devices:

```
[edit]
user@host# set applications application customer-msrpc application-protocol ms-rpc
user@host# set applications application customer-msrpc term t1 protocol tcp
```

```
user@host# set applications application customer-msrpc term t1 uuid e3514235-4b06-11d1-
ab04-00c04fc2dcd2
```

MS-RPC data traffic with TCP, uuid e3514235-4b06-11d1-ab04-00c04fc2dcd2, will be permitted, when custom-msrpc is applied to the policy along with other predefined junos-ms-rpc** applications.

SEE ALSO

[Understanding RPC ALGs](#)

Understanding the IPv6 DNS ALG for Routing, NAT, and NAT-PT

IN THIS SECTION

- [IPv6 DNS ALG Traffic in NAT mode | 5](#)
- [IPv6 DNS ALG Traffic in NAT-PT mode | 6](#)

Domain Name System (DNS) is the part of the ALG that handles DNS traffic, monitors DNS query and reply packets, and closes the session if the DNS flag indicates the packet is a reply message.

The DNS ALG supports IPv4 in route mode for Junos OS Release 10.0 and earlier releases. In Junos OS Release 10.4, this feature implements IPv6 support on the DNS ALG for routing, Network Address Translation (NAT), and Network Address Translation-Protocol Translation (NAT-PT).

When the DNS ALG receives a DNS query from the DNS client, a security check is done on the DNS packet. When the DNS ALG receives a DNS reply from the DNS server, a similar security check is done, and then the session for the DNS traffic closes.

IPv6 DNS ALG Traffic in NAT mode

IPv6 NAT provides address translation between IPv4 and IPv6 addressed network devices. It also provides address translation between IPv6 hosts. NAT between IPv6 hosts is done in a similar manner and for similar purposes as IPv4 NAT.

When the DNS traffic works in NAT mode, the DNS ALG translates the public address in a DNS reply to a private address when the DNS client is on private network, and similarly translates a private address to a public address when the DNS client is on a public network.

In Junos OS Release 10.4 IPv6 NAT supports:

- Source NAT translations
- Destination NAT mappings
- Static NAT mappings



NOTE: The IPv6 DNS ALG NAT supports only static NAT mapping.

IPv6 DNS ALG Traffic in NAT-PT mode

IPv6 NAT-PT provides address allocation and protocol translation between IPv4 and IPv6 addressed network devices. The translation process is based on the Stateless IP/ICMP Translation (SIIT) method; however, the state and the context of each communication is retained during the session lifetime. IPv6 NAT-PT supports Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) packets.

IPv6 NAT-PT supports the following types of NAT-PT:

- Traditional NAT-PT
- Bidirectional NAT-PT

A DNS-based mechanism dynamically maps IPv6 addresses to IPv4-only servers. NAT-PT uses the DNS ALG to transparently do the translations.

For example, a company using an internal IPv6 network needs to be able to communicate with external IPv4 servers that do not have IPv6 addresses.

To support the dynamic address binding, a DNS should be used for name resolution. The IPv4 host looks up the name of the IPv6 node in its local configured IPv4 DNS server, which then passes the query to the IPv6 DNS server through the device using NAT-PT.

When DNS traffic works in NAT-PT mode, the DNS ALG translates the IP address in a DNS reply packet between the IPv4 address and the IPv6 address when the DNS client is in an IPv6 network and the server is in an IPv4 network, and vice versa.



NOTE: In NAT-PT mode, only IPV4 to IPV6 addresses translation is supported in the DNS ALG. To support NAT-PT mode in a DNS ALG, the NAT module should support NAT-PT.

When the DNS ALG receives a DNS query from the DNS client, the DNS ALG performs the following security and sanity checks on the DNS packets:

- Enforces the maximum DNS message length (the default is 512 bytes and the maximum length is 8KB)
- Enforces a domain-name length of 255 bytes and a label length of 63 bytes
- Verifies the integrity of the domain-name referred to by the pointer if compression pointers are encountered in the DNS message
- Checks to see if a compression pointer loop exists

Similar sanity checks are done when the DNS ALG receives a DNS reply from the DNS Server, after which the session for this DNS traffic gets closed.

SEE ALSO

[DNS ALG Overview](#)

[IPv6 NAT Overview](#)

[IPv6 NAT PT Overview](#)

Understanding IPv6 Support in FTP ALG

IN THIS SECTION

- [FTP ALG Support for IPv6 | 8](#)
- [EPRT mode | 8](#)
- [EPSV mode | 8](#)

File Transfer Protocol (FTP) is the part of the ALG that handles FTP traffic. The PORT/PASV requests and corresponding 200/227 responses in FTP are used to announce the TCP port, which the host listens to for the FTP data connection.

EPRT/EPSV/229 commands are used for these requests and responses. FTP ALG supports EPRT/EPSV/229 already, but only for IPv4 addresses.

In Junos OS Release 10.4, EPRT/EPSV/229 commands have been updated to support both IPv4 and IPv6 addresses.

FTP ALG uses preallocated objcache to store its session cookies. When both IPv4 and IPv6 addresses are supported on FTP ALG, the session cookie structure will enlarge by 256 bits (32 bytes) to store IPv6 address.

FTP ALG Support for IPv6

The FTP ALG monitors commands and responses on the FTP control channel for syntactical correctness and opens corresponding pinholes to permit data channel connections to be established. In Junos OS Release 10.4, the FTP ALG supported IPv4 routing, IPv6 routing, and NAT mode only. In Junos OS Release 11.2 and later releases, the FTP ALG also supports IPv6 NAT and NAT-PT modes.

EPRT mode

The EPRT command allows for the specification of an extended address for the data connection. The extended address must consist of the network protocol as well as the network and transport addresses.

The format of EPRT is:

EPRT<space><d><net-prt><d><net-addr><d><tcp-port><d>

- <net-prt>: An address family number defined by IANA
- <net-addr>: A protocol specific string of the network address
- <tcp-port>: A TCP port number

The following are sample EPRT commands for IPv6:

EPRT |2|1080::8:800:200C:417A|5282|

In this mode, FTP ALG focuses only on the EPRT command; it extracts the IPv6 address and port from the EPRT command and opens the pinhole.

EPSV mode

The EPSV command requests that a server be listening on a data port and waiting for a connection. The response to this command includes only the TCP port number of the listening connection.

An example response string is follows:

```
Entering Extended Passive Mode (|||6446|)
```



NOTE: The response code for entering passive mode using an extended address must be 229. You should extract the TCP port in 229 payloads and use it to open the pinhole.

SEE ALSO

| [FTP ALG Overview](#) | 28

Understanding TAP Mode Support for ALG

The Terminal Access Point (TAP) mode is a standby device, which checks the mirrored traffic through switch. The TAP mode does not depend on ALG enabled or disabled status. The ALG configuration remains the same as non-TAP mode.

When you configure an SRX Series Firewall to operate in TAP mode, the device generates security log information to display the information on threats detected, application usage, and user details. When the device is configured to operate in TAP mode, the device receives packets only from the configured TAP interface. Except the configured TAP interface, other interfaces are configured to normal interface that is used as management interface or connected to the outside server. The SRX Series Firewall will generate security report or log according to the incoming traffic.

ALG supports the application such as payload NAT, and dynamically permit its data traffic.



NOTE: You can configure only one TAP interface when you operate the device in TAP mode.

Enabling and Disabling ALG in TAP Mode

This topic shows how to enable or disable the ALG status in TAP mode.

Before you begin:

Read the [Understanding TAP Mode Support for ALG](#) to understand about ALG support for TAP mode.

- The default ALG status for SRX300, SRX320, SRX340, SRX345, SRX380, and SRX550M devices is as follows:

```
ALG Status:
DNS       : Enabled
FTP       : Enabled
H323      : Enabled
MGCP      : Enabled
MSRPC     : Enabled
PPTP      : Enabled
RSH       : Disabled
RTSP      : Enabled
SCCP      : Enabled
SIP       : Enabled
SQL       : Disabled
SUNRPC    : Enabled
TALK      : Enabled
TFTP      : Enabled
IKE-ESP   : Disabled
TWAMP     : Disabled
```

- The default ALG status for SRX4100 device is as follows:

```
ALG Status:
DNS       : Enabled
FTP       : Enabled
H323      : Disabled
MGCP      : Disabled
MSRPC     : Enabled
PPTP      : Enabled
RSH       : Disabled
RTSP      : Disabled
SCCP      : Disabled
SIP       : Disabled
SQL       : Disabled
SUNRPC    : Enabled
TALK      : Enabled
TFTP      : Enabled
IKE-ESP   : Disabled
TWAMP     : Disabled
```

- To enable the ALG that is disabled by default, use the following command.

```
[edit]  
user@host# set security alg alg-name
```

To change back the enabled ALG to the default status, use the following command.

```
[edit]  
user@host# delete security alg alg-name
```

- To disable ALG that is enabled by default, use the following command.

```
[edit]  
user@host# set security alg alg-name disable
```

To change back the disabled ALG to the default status, use the following command.

```
[edit]  
user@host# delete security alg alg-name disable
```

- To enable the IKE ALG, use the following command.

```
[edit]  
user@host# set security alg ike-esp-nat enable
```

To change back the enabled IKE ALG to the default status, use the following command.

```
[edit]  
user@host# delete security alg ike-esp-nat enable
```

RELATED DOCUMENTATION

[Understanding Data ALG Types | 13](#)

[Understanding VoIP ALG Types | 188](#)

2

CHAPTER

Data ALGs

IN THIS CHAPTER

- Understanding Data ALG Types | 13
 - DNS ALG | 15
 - FTP ALG | 28
 - IKE and ESP ALG | 37
 - PPTP ALG | 53
 - RPC ALG | 72
 - RSH ALG | 87
 - RTSP ALG | 106
 - SQLNET ALG | 124
 - TALK ALG | 144
 - TFTP ALG | 163
 - TWAMP ALG | 172
 - Understanding IPv6 ALG Support for ICMP | 176
 - Understanding 464XLAT ALG Traffic Support | 178
 - Understanding ALG Support for VRF Routing Instance | 186
-

Understanding Data ALG Types

Junos OS supports the data ALG types listed in [Table 1 on page 13](#).

Table 1: Data ALG Types

Data ALG	Description
DNS	Provides an ALG for the Domain Name System. The DNS ALG monitors DNS query and reply packets and closes session if the DNS flag indicates the packet is a reply message.
DDNS	Dynamic DNS (DDNS) is an addition to the DNS standard. DDNS updates a DNS server with new or changed records for IP addresses without the need for human intervention. Unlike DNS that only works with static IP addresses, DDNS is also designed to support dynamic IP addresses, such as those assigned by a DHCP server. DDNS is a good option for home networks, which often receive dynamic public IP addresses from their Internet provider that occasionally changes.
FTP	Provides an ALG for the File Transfer Protocol (FTP).The FTP ALG monitors PORT, PASV, and 227 commands. It performs NAT on the IP, port, or both in the message and gate opening on the device as necessary.
IKE and ESP ALG	<p>Monitors IKE traffic between the client and the server and permits only one IKE Phase 2 message exchange between any given client/server pair, not just one exchange between any client and any server.</p> <p>Internet Key Exchange (IKE) and Encapsulating Security Payload (ESP) traffic is exchanged between the clients and the server. However, if the clients do not support NAT-Traversal (NAT-T) and if the device assigns the same NAT-generated IP address to two or more clients, the device will be unable to distinguish and route return traffic properly.</p> <p>NOTE: If the user wants to support both NAT-T-capable and non-NAT-T-capable clients, then some additional configurations are required. If there are NAT-T capable clients, the user must enable the source NAT address persistence.</p>
MS-RPC	Provides an ALG for the Microsoft Remote Procedure Call.

Table 1: Data ALG Types (Continued)

Data ALG	Description
PPTP	Provides an ALG for the Point-to-Point Tunneling Protocol (PPTP). The PPTP is a Layer 2 protocol that tunnels PPP data across TCP/IP networks. The PPTP client is freely available on Windows systems and it is also popularly applied on Linux systems, and is widely deployed for building Virtual Private Networks (VPNs).
RSH	Provides an ALG for the Remote Shell (RSH). The RSH ALG handles TCP packets destined for port 514 and processes the RSH port command. The RSH ALG performs NAT on the port in the port command and opens gates as necessary.
RTSP	Provides an ALG for the Real Time Streaming Protocol (RTSP). RTSP is a standard protocol for streaming media applications. It controls the delivery of data with real-time properties such as audio and video.
SQL	Provides an ALG for the Structured Query Language (SQL). The SQLNET ALG processes SQL TNS response frame from the server side. It parses the packet and looks for the (HOST=ipaddress), (PORT=port) pattern and performs NAT and gate opening on the client side for the TCP data channel.
SUNRPC	Provides an ALG for the SUN Remote Procedure Call.
TALK	Provides an ALG for the TALK Protocol. The TALK protocol uses UDP port 517 and port 518 for control channel connections. The talk program consists of a server and a client. The server handles client notifications and helps to establish talk sessions. There are two types of talk servers: ntalk and talkd. The TALK ALG processes packets of both ntalk and talkd formats. It also performs NAT and gate opening as necessary.
TFTP	Provides an ALG for the Trivial File Transfer Protocol (TFTP). The TFTP ALG processes TFTP packets that initiate the request and opens a gate to allow return packets from the reverse direction to the port that sends the request.
TWAMP	The Two-Way Active Measurement Protocol (TWAMP) is an open protocol for measuring network performance between any two devices in a network that supports the protocols in the TWAMP framework. The TWAMP consists of two interrelated protocols – TWAMP-Control and TWAMP-Test.

For information about enabling and configuring each of these ALGs through J-Web, select the **Configure>Security>ALG** page in the J-Web user interface and click **Help**.

The following ALG data traffic supports Express Path—FTP, H.323 (only RTP/RTCP sessions are offloaded), MGCP, MS RPC, RSH, RTSP, SCCP, SIP (only RTP/RTCP sessions are offloaded), Sun RPC, TALK (only TCP sessions are offloaded), and TFTP.

DNS, IKE and ESP, PPTP, and SQL-NET ALG data traffic do not support Express Path.

Once an Express Path session is set up, packets cannot be sent to the SPU again.

RELATED DOCUMENTATION

| [Understanding Custom ALG Services](#) | 3

DNS ALG

IN THIS SECTION

- [DNS ALG Overview](#) | 16
- [Example: Configuring the DNS ALG](#) | 17
- [Understanding DNS and DDNS Doctoring](#) | 23
- [Platform-Specific DNS ALG Behavior](#) | 27

The Domain Name System (DNS) Application Layer Gateway (ALG) service 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.

Use [Feature Explorer](#) to confirm platform and release support for specific features.

Review the "[Platform-Specific DNS ALG Behavior](#)" on page 27 section for notes related to your platform.

DNS ALG Overview

The DNS Application Layer Gateway (ALG) service provides an application-level gateway for use with DNS clients. The DNS ALG service allows a client to access multiple DNS servers in different networks and provides routing to and from those servers. It also supports flexible address translation of the DNS query and response packets. These functions allow the DNS client to query many different domains from a single DNS server instance on the client side of the network.

The DNS server listens through UDP port 53 for incoming queries from DNS resolvers. A resolver communicates with DNS servers by sending DNS queries and handling DNS responses.



NOTE: The default port for DNS ALG is port 53.

The DNS ALG performs the following functions:

- Monitors DNS query and reply packets and closes the session when the DNS reply is received
- Performs DNS doctoring
- Performs the IPv4 and IPv6 address transformations

The Domain Name System (DNS) was originally designed to support queries of a static configured database and the data was expected to change.

Dynamic DNS (DDNS) support is now available in addition to the DNS standard. The main difference between DNS and DDNS is in the message format of the header section and the update message.

DDNS messages are processed differently when compared to DNS messages. Message parsing is rewritten for DDNS. DDNS does NAT and NAT-PT in the query part of the message and DNS does NAT and NAT-PT in the response part of the message.

SEE ALSO

[DNS Overview](#)

[DNSSEC Overview](#)

Example: Configuring the DNS ALG

IN THIS SECTION

- [Requirements | 17](#)
- [Overview | 17](#)
- [Configuration | 18](#)
- [Verification | 21](#)

This example shows how to configure the DNS ALG to pass through DNS traffic with a static NAT pool on Juniper Networks devices.

Requirements

Before you begin:

- Configure static NAT pool for all IP address.
- Understand the concepts behind ALG for DNS. See ["DNS ALG Overview" on page 16](#).

Overview

IN THIS SECTION

- [Topology | 17](#)

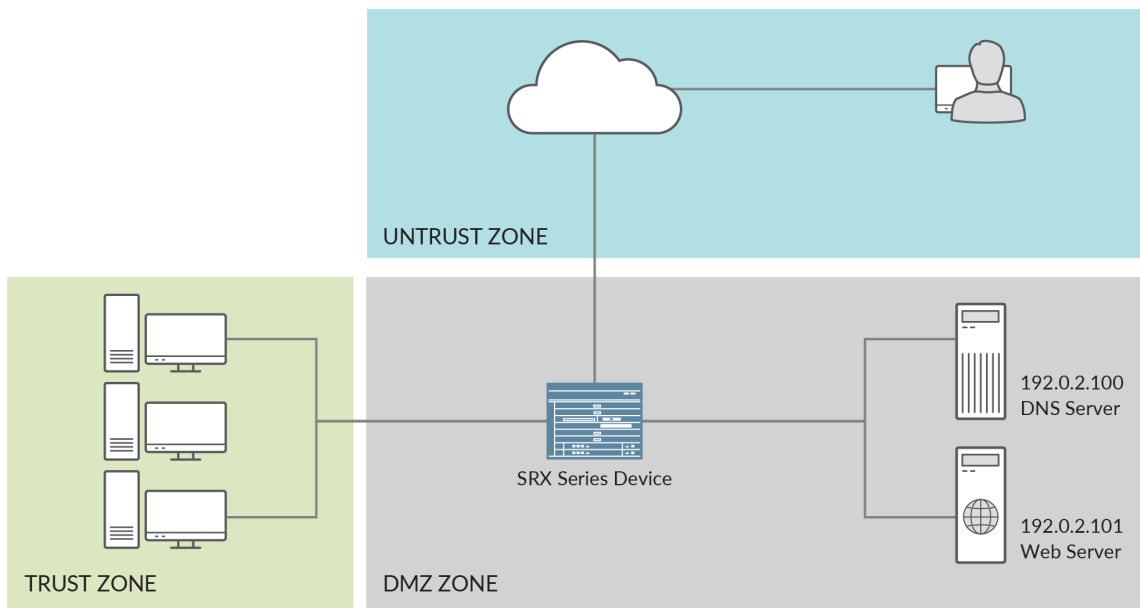
In this example, the ALG for DNS is configured to monitor and allow DNS traffic to be exchanged between the clients and the server located on opposite sides of a Juniper Networks device.

This example shows how to configure a static NAT pool and rule set, and associate the DNS ALG to a policy.

Topology

[Figure 1 on page 18](#) shows the DNS ALG topology.

Figure 1: DNS ALG Topology



Static NAT Host	Static Private IP	Public IP
DNS Server	192.0.2.100	203.0.113.100
Web Server	192.0.2.101	203.0.113.101

g300040

Configuration

IN THIS SECTION

- [Configuring a NAT Static Pool and Rule Set | 18](#)
- [Configuring and Printing the DNS Trace | 21](#)

Configuring a NAT Static Pool and Rule Set

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy

and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security nat static rule-set rs1 from zone untrust
set security nat static rule-set rs1 rule r1 match destination-address 203.0.113.100
set security nat static rule-set rs1 rule r1 then static-nat prefix 192.0.2.100
set security policies from-zone untrust to-zone trust policy u2t match source-address any
set security policies from-zone untrust to-zone trust policy u2t match destination-address any
set security policies from-zone untrust to-zone trust policy u2t match application junos-dns-udp
set security policies from-zone untrust to-zone trust policy u2t then permit
```

Step-by-Step Procedure

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

To configure a static NAT pool:

1. Create a NAT static rule set.

```
[edit]
user@host# set security nat static rule-set rs1 from zone untrust
user@host# set security nat static rule-set rs1 rule r1 match destination-address
203.0.113.100
user@host# set security nat static rule-set rs1 rule r1 then static-nat prefix 192.0.2.100
```

2. Associate the DNS application using a policy.

```
[edit]
user@host# set security policies from-zone untrust to-zone trust policy u2t match source-
address any
user@host# set security policies from-zone untrust to-zone trust policy u2t match destination-
address any
user@host# set security policies from-zone untrust to-zone trust policy u2t match application
junos-dns-udp
user@host# set security policies from-zone untrust to-zone trust policy u2t then permit
```


Results

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

```
user@host# show security nat
static {
  rule-set rs1 {
    from zone untrust;
    rule r1 {
      match {
        destination-address 203.0.113.100;
      }
      then {
        static-nat {
          prefix {
            192.0.2.100;
          }
        }
      }
    }
  }
}
```

```
[edit]
user@host# show security policies
from-zone untrust to-zone trust {
  policy u2t {
    match {
      source-address any;
      destination-address any;
      application [ junos-dns-udp];
    }
    then {
      permit;
    }
  }
}
default-policy {
```

```
    permit-all;  
}
```

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

Configuring and Printing the DNS Trace

Purpose

Print the DNS trace file.

Action

From configuration mode, enter the following command.

```
set security alg traceoptions file alglog  
set security alg traceoptions file size 1g  
set security alg traceoptions level verbose  
set security alg dns traceoptions flag all
```

Verification

IN THIS SECTION

- [Verifying DNS ALG | 21](#)
- [Verifying DNS ALG Security Flow Session | 22](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying DNS ALG

Purpose

Verify that DNS ALG is enabled.

Action

From operational mode, enter the `show security alg status` command.

```
user@host> show security alg status
```

ALG Status :

DNS	: Enabled
FTP	: Enabled
H323	: Disabled
MGCP	: Disabled
MSRPC	: Enabled
PPTP	: Enabled
RSH	: Disabled
RTSP	: Disabled
SCCP	: Disabled
SIP	: Disabled
SQL	: Disabled
SUNRPC	: Enabled
TALK	: Enabled
TFTP	: Enabled
IKE-ESP	: Disabled

Meaning

The output shows the DNS ALG is enabled.

Verifying DNS ALG Security Flow Session

Purpose

Verify ALG security flow session is enabled.

Action

From operational mode, enter the `show security flow session application dns extensive` command.

```
user@host> show security flow session application dns extensive
```

Session ID: 24088, Status: Normal

Flags: 0x40/0x0/0x2/0x2000103

Policy name: unt2tru/6

```

Source NAT pool: Null, Application: junos-dns-udp/16
Dynamic application: junos:UNKNOWN,
Encryption: Unknown
Application traffic control rule-set: INVALID, Rule: INVALID
Maximum timeout: 60, Current timeout: 56
Session State: Valid
Start time: 658866, Duration: 10
  In: 192.0.2.0/38926 --> 198.51.100.0/53;udp,
  Conn Tag: 0x0, Interface: ge-0/0/3.0,
    Session token: 0xa, Flag: 0x621
    Route: 0x110010, Gateway: 192.0.2.0, Tunnel: 0
    Port sequence: 0, FIN sequence: 0,
    FIN state: 0,
    Pkts: 2, Bytes: 116
  Out: 198.51.100.0/53 --> 192.0.2.0/38926;udp,
  Conn Tag: 0x0, Interface: ge-0/0/2.0,
    Session token: 0x9, Flag: 0x620
    Route: 0x100010, Gateway: 198.51.100.0, Tunnel: 0
    Port sequence: 0, FIN sequence: 0,

```

Meaning

The output shows there is an active flow utilizing the DNS ALG.

SEE ALSO

[DNS ALG | 15](#)

Understanding DNS and DDNS Doctoring

IN THIS SECTION

- [Disabling DNS and DDNS Doctoring | 27](#)

Junos OS supports Domain Name System (DNS) for ALGs. The DNS ALG monitors DNS query and reply packets and closes the session if the DNS flag indicates that the packet is a reply message. To configure the DNS ALG, use the `edit security alg dns` statement at the `[edit security alg]` hierarchy level.

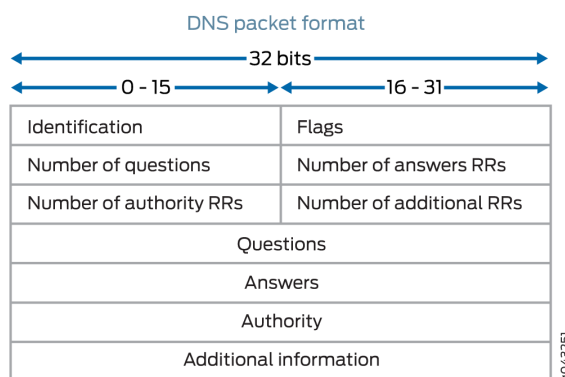
DNS provides name-to-address mapping within a routing class, whereas Network Address Translation (NAT) attempts to provide transparent routing between hosts in disparate address realms of the same routing class. As a result, NAT can cause some DNS problems the DNG ALG must handle through a process called *DNS* doctoring.

The same doctoring feature applies to the dynamic domain name system (DDNS). For DDNS in NAT mode, you also can do the IP translation in the DDNS update.

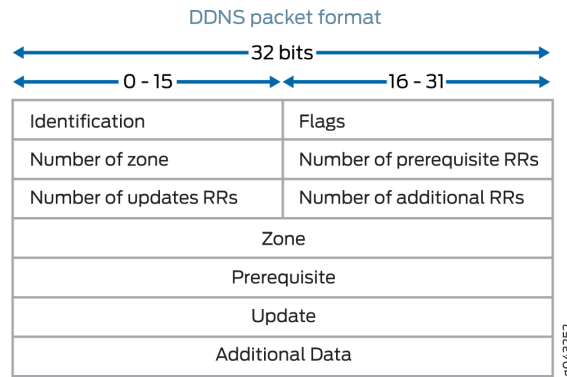
To resolve the problems introduced by NAT, DNS and DDNS ALG functionality has been extended to support static NAT, allowing the problems to be resolved through DNS doctoring.

The restoring and doctoring process is performed in two parts:

- **Packet sanity check**



For the DNS packet, the DNS ALG check fields are questions, answers, authority, and additional information. The DNS ALG drops the packet if the number of questions is more than 1, the domain name is more than 255 bytes, or the label length is more than 63 bytes.



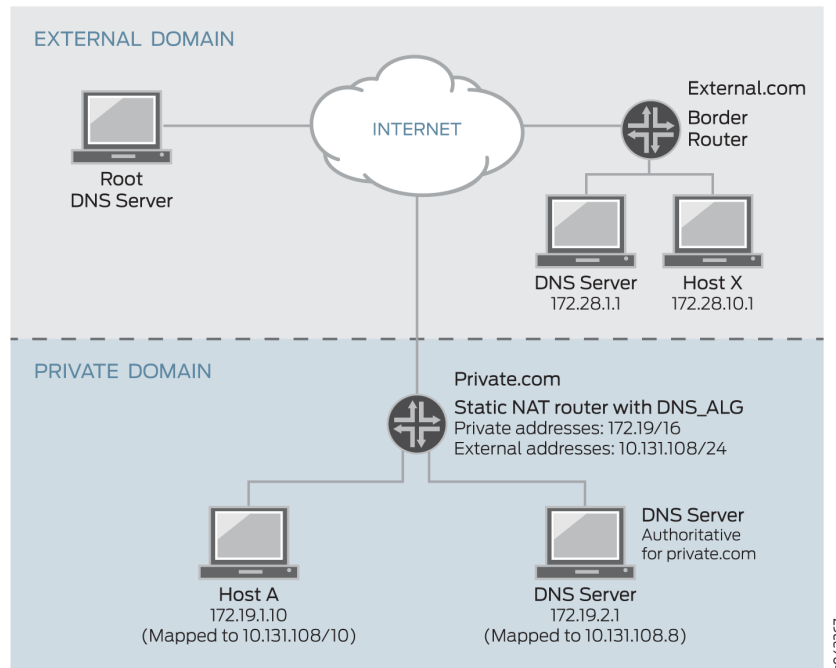
For the DDNS packet, the DNS ALG check fields are zone, prerequisite, update, and additional data. The DNS ALG drops the packet if the number of zones is more than 1, the domain name is more than 255 bytes, or the label length is more than 63 bytes.

For both DNS and DDNS, the DNS ALG drops the packet that does not comply with the standards.

- NAT

Figure 2 on page 25 shows how DNS translates a private address to a public address.

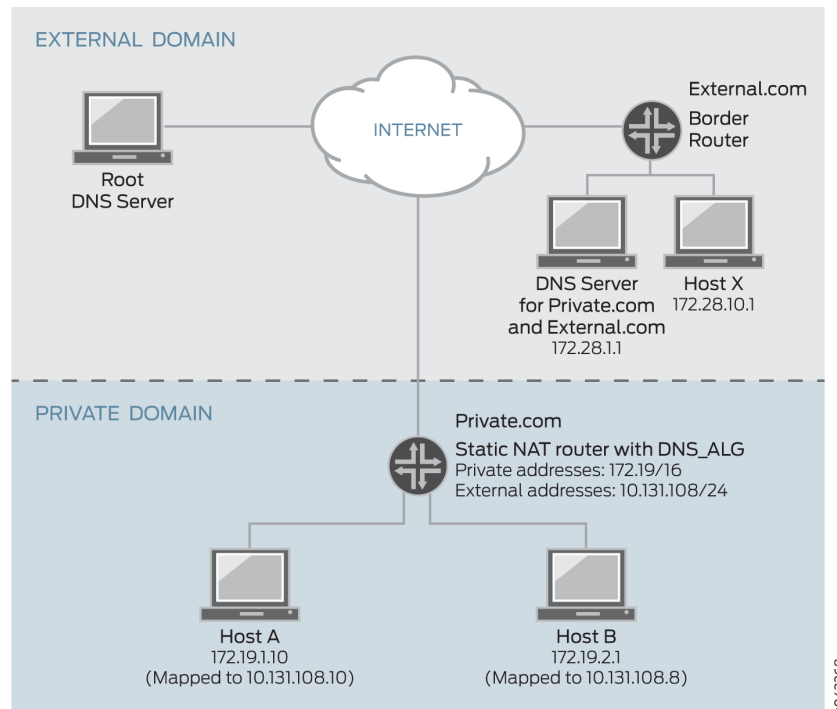
Figure 2: DNS Address Translation (Private to Public)



When host X in external.com wants to resolve host A's address through DNS and if the DNS ALG does not support NAT, it takes a private address such as 172.19.1.10, which is invalid to host X. The private address is translated to public address 10.131.108.10 through the DNS ALG.

Figure 3 on page 26 shows how DNS translates a public address to a private address.

Figure 3: DNS Address Translation (Public to Private)



When host A in private.com wants to resolve host B's address through DNS and if the DNS ALG does not support NAT, it takes a public address from the DNS server in external.com, such as 10.131.108.8. If Host A sends traffic to host B with public address 10.131.108.8, which is invalid to host B in the private domain. Hence, the public address in the DNS query A-record is translated to private address 172.19.2.1 through the DNS ALG.



NOTE: The DNS ALG can translate the first 32 A-records in a single DNS reply. A-records after the first 32 records are not handled. Also note that the DNS ALG supports IPv4 and IPv6 addresses and does not support VPN tunnels.

Disabling DNS and DDNS Doctoring

The DNS ALG must be enabled on the devices to perform DNS and DDNS doctoring. With the DNS ALG enabled on the device, the DNS and DDNS doctoring feature is enabled by default. You can disable DNS and DDNS doctoring with the CLI.

To disable DNS and DDNS doctoring:

1. Disable all the doctoring features by specifying the `none` configuration option.

This command disables all the doctoring features.

```
user@host# set security alg dns doctoring none
```

2. Disable the NAT feature and retain the sanity-check feature by specifying the `sanity-check` configuration option.

This option disables the NAT feature and retains the sanity-check feature.

```
user@host# set security alg dns doctoring sanity-check
```

3. If you are finished configuring the device, commit the configuration.
4. To verify the configuration, use the vty command **show usp algs dns stats**.

Platform-Specific DNS ALG Behavior

Use [Feature Explorer](#) to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platform:

Platform	Difference
SRX Series	<ul style="list-style-type: none"> SRX3400, SRX3600, SRX4600, SRX5600, and SRX5800 devices that support DNS ALG enable DNS doctoring by default.

RELATED DOCUMENTATION

[IKE and ESP ALG | 37](#)[PPTP ALG | 53](#)[RPC ALG | 72](#)

FTP ALG

IN THIS SECTION

- [FTP ALG Overview | 28](#)
- [Understanding FTP Commands | 30](#)
- [Example: Configuring the FTP ALG | 32](#)

File Transfer Protocol is a widely and commonly used method of exchanging files over IP networks. The FTP ALG monitors PORT, PASV, and 227 commands. It performs NAT on the IP, port, or both in the message and gate opening on the device as necessary.

FTP ALG Overview

IN THIS SECTION

- [IPv6 FTP ALG for Routing | 29](#)

The File Transfer Protocol (FTP) is a widely and commonly used method of exchanging files over IP networks. 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 active 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.

FTP represents the 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.

The FTP ALG supports the following:

- Automatically allocates data ports and firewall permissions for dynamic data connection
- Monitors the control connection in both active and passive modes
- Rewrites the control packets with the appropriate NAT address and port information
- Network Address Translation, Protocol Translation (NAT-PT)
- Transport Layer Security (TLS) as the security mechanism

IPv6 FTP ALG for Routing

The PORT/PASV requests and corresponding 200/227 responses in FTP are used to announce the TCP port, which the host listens to for the FTP data connection.

EPRT/EPSV/229 commands are used for these requests and responses. FTP ALG supports EPRT/EPSV/229 already, but only for IPv4 addresses.

In Junos OS Release 10.4, EPRT/EPSV/229 commands have been updated to support both IPv4 and IPv6 addresses.

FTP ALG uses preallocated objcache to store its session cookies. When both IPv4 and IPv6 addresses are supported on FTP ALG, the session cookie structure will enlarge by 256 bits (32 bytes) to store IPv6 address.

FTP ALG Support for IPv6

The FTP ALG monitors commands and responses on the FTP control channel for syntactical correctness and opens corresponding pinholes to permit data channel connections to be established. In Junos OS Release 10.4, the FTP ALG supported IPv4 routing, IPv6 routing, and NAT mode only. In Junos OS Release 11.2 and later releases, the FTP ALG also supports IPv6 NAT and NAT-PT modes..

Understanding FTP Commands

IN THIS SECTION

- [PORT Command | 30](#)
- [PASV Command | 30](#)
- [Extended FTP Commands | 30](#)

The FTP ALG monitors commands and responses on the FTP control channel for syntactical correctness and opens corresponding pinholes to permit data channel connections to be established. In Junos OS Release 10.4, the FTP ALG supported IPv4 routing and NAT mode, and IPv6 routing mode only. In Junos OS Release 11.2 and later releases, the FTP ALG also supports IPv6 NAT and NAT-PT modes.

PORT Command

The PORT command is used in active FTP mode. The PORT command specifies the address and the port number to which a server should connect. When you use this command, the argument is a concatenation of a 32-bit Internet host address and a 16-bit TCP port address. The address information is broken into 8-bit fields, and the value of each field is transmitted as a decimal number (in character string representation). The fields are separated by commas.

The following is a sample PORT command, where h1 is the highest order 8-bit of the Internet host address:

```
PORT h1,h2,h3,h4,p1,p2
```

PASV Command

The PASV command requests a server to listen on a data port that is not the default data port of the server and to wait for a connection, rather than initiating another connection. The response to the PASV command includes the host and port address the server is listening on.

Extended FTP Commands

Extended FTP commands provide a method by which FTP can communicate the data connection endpoint information for network protocols other than IPv4. Extended FTP commands are specified in RFC 2428. In RFC 2428, the extended FTP commands EPRT and EPSV, replace the FTP commands PORT and PASV, respectively.

EPRT Command

The EPRT command allows for the specification of an extended address for the data connection. The extended address must consist of the network protocol as well as the network and transport addresses.

The format of EPRT is:

EPRT<space><d><net-prt><d><net-addr><d><tcp-port><d>

Parameter	Description
net-prt	An address family number defined by IANA.
net-addr	A protocol-specific string of the network address.
tcp-port	A TCP port number on which the host is listening for data connection.
Delimiter	The delimiter character must be one of the ASCII characters in range 33 to 126 inclusive. The character " " (ASCII 124) is recommended.

The following command shows how to specify the server to use an IPv4 address to open a data connection to host 132.235.1.2 on TCP port 6275:

```
EPRT |1|132.235.1.2|6275|
```

The following command shows how to specify the server to use an IPv6 network protocol and a network address to open a TCP data connection on port 5282:

```
EPRT |2|1080::8:800:200C:417A|5282|
```

In this mode, FTP ALG focuses only on the EPRT command; it extracts the IPv6 address and port from the EPRT command and opens the pinhole.

EPSV mode

The EPSV command requests that a server listen on a data port and wait for a connection. The response to this command includes only the TCP port number of the listening connection.

An example response string is as follows:

```
Entering Extended Passive Mode (||6446|)
```



NOTE: The response code for entering passive mode using an extended address must be 229. You should extract the TCP port in 229 payloads and use it to open the pinhole.

Example: Configuring the FTP ALG

IN THIS SECTION

- [Requirements | 32](#)
- [Overview | 32](#)
- [Configuration | 32](#)
- [Verification | 36](#)

This example shows how to configure the NAT-PT for FTP ALG.

Requirements

Before you begin:

- Configure proxy ARP for all IP addresses in the source NAT pool.
- Understand the concepts behind ALG for FTP. See ["FTP ALG Overview" on page 28](#).

Overview

In this example, the ALG for FTP is configured to monitor and allow FTP traffic to be exchanged between the clients and the server located on opposite sides of a Juniper Networks device.

This example shows how to configure the NAT-PT for FTP ALG.

Configuration

IN THIS SECTION

- [Configuring a NAT Source Pool, NAT Static Pool and Rule Set | 33](#)

Configuring a NAT Source Pool, NAT Static Pool and Rule Set

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security nat static rule-set rs1 from zone untrust
set security nat source rule-set rs-source to zone trust
set security nat source rule-set rs-source rule src-nat match source-address 3333::130/128
set security nat source rule-set rs-source rule src-nat match destination-address 40.0.0.211/32
set security nat source rule-set rs-source rule src-nat then source-nat interface
set security nat static rule-set rs2 from zone untrust
set security nat static rule-set rs2 rule r2 match destination-address 4444::141/128
set security nat static rule-set rs2 rule r2 then static-nat prefix 40.0.0.211/32
```

Step-by-Step Procedure

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

To configure a source NAT pool:

1. Create a source NAT, static NAT, and interface NAT rule set.

```
[edit ]
user@host# set security nat source rule-set rs-source from zone untrust
user@host# set security nat source rule-set rs-source to zone trust
user@host# set security nat source rule-set rs-source rule src-nat match source-address
3333::130/128
user@host# set security nat source rule-set rs-source rule src-nat match destination-address
40.0.0.211/32
user@host# set security nat source rule-set rs-source rule src-nat then source-nat interface
user@host# set security nat static rule-set rs2 from zone untrust
```

```

user@host# set security nat static rule-set rs2 rule r2 match destination-address
4444::141/128
user@host# set security nat static rule-set rs2 rule r2 then static-nat prefix 40.0.0.211/32

```

2. Associate the NAT-PT application using a policy.

```

[edit]
user@host# set security policies from-zone trust to-zone untrust policy ftp-basic match
source-address any
user@host# set security policies from-zone trust to-zone untrust policy ftp-basic match
destination-address any
user@host# set security policies from-zone trust to-zone untrust policy ftp-basic match
application junos-ftp
user@host# set security policies from-zone trust to-zone untrust policy ftp-basic then permit

```

Results

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

```

user@host# show security nat
static {
    rule-set rs2 {
        from zone untrust;
        rule r2 {
            match {
                destination-address 4444::141/128;
            }
            then {
                static-nat {
                    prefix {
                        40.0.0.211/32
                    }
                }
            }
        }
    }
}

```

```
    }
}
```

```
[edit]
user@host# show security policies
from-zone untrust to-zone trust {
  policy ftp-basic {
    match {
      source-address any;
      destination-address any;
      application [ junos-ping junos-mgcp junos-ftp junos-rsh junos-h323 ];
    }
    then {
      permit;
    }
  }
}
default-policy {
  permit-all;
}
```

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

Configuring FTP ALG Security Extension

Purpose

Set the security alg ftp extension

Action

From configuration mode, enter the following command.

```
set security alg ftp ftps-extension
```


Verification

IN THIS SECTION

- [Verifying the NAT Source Pool, NAT Static Pool Rule Set | 36](#)
- [Verifying FTP ALGs | 36](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying the NAT Source Pool, NAT Static Pool Rule Set

Purpose

Verify that the NAT source pool and rule set used to support the FTP ALG are working properly.

Action

From operational mode, enter the `show configuration security nat` command.

Verifying FTP ALGs

Purpose

Verify that FTP ALG is enabled.

Action

From the operational mode, enter the `show security alg status` command.


```
user@host> show security alg status
FTP      : Enabled
```

Meaning

The output shows the FTP ALG status as follows:

- Enabled—Shows the FTP ALG is enabled.

- Disabled—Shows the FTP ALG is disabled.

 **NOTE:** The FTP ALG is enabled by default.

Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
11.2	In Junos OS Release 11.2 and later releases, the FTP ALG also supports IPv6 NAT and NAT-PT modes.
10.4	In Junos OS Release 10.4, EPRT/EPSV/229 commands have been updated to support both IPv4 and IPv6 addresses.
10.4	In Junos OS Release 10.4, the FTP ALG supported IPv4 routing, IPv6 routing, and NAT mode only.

RELATED DOCUMENTATION

PPTP ALG 53
RPC ALG 72

IKE and ESP ALG

IN THIS SECTION

- [Understanding the IKE and ESP ALG | 38](#)
- [Example: Configuring the IKE and ESP ALG | 40](#)
- [Example: Enabling the IKE and ESP ALG and Setting Timeouts | 49](#)
- [Platform-Specific IKE ALG Behavior | 52](#)

Internet Key Exchange (IKE) and Encapsulating Security Payload (ESP) are a part of the IP Security (IPsec) protocol. IKE and ESP traffic is exchanged between the clients and the server. The IKE and ESP

ALG helps in resolving the IPsec VPNs issues when the IPsec VPN passes through the device of which NAT is enabled.

Use [Feature Explorer](#) to confirm platform and release support for specific features.

Review the "[Platform-Specific IKE ALG Behavior](#)" on [page 52](#) section for notes related to your platform.

Understanding the IKE and ESP ALG

IN THIS SECTION

- [Understanding IKE and ESP ALG Operation | 39](#)

A Network Address Translation (NAT) device when placed between VPN clients on the private side of the NAT gateway and the virtual private network (VPN) gateways on the public side.

Internet Key Exchange (IKE) and Encapsulating Security Payload (ESP) traffic is exchanged between the clients and the server. However, if the clients do not support NAT-Traversal (NAT-T) and if the device assigns the same NAT-generated IP address to two or more clients, the device will be unable to distinguish and route return traffic properly.



NOTE: If the user wants to support both NAT-T-capable and non-NAT-T-capable clients, then some additional configurations are required. If there are NAT-T capable clients, the user must enable the source NAT address persistence.

The ALG for IKE and ESP monitors IKE traffic between the client and the server and permits only one IKE Phase 2 message exchange between any given client/server pair, not just one exchange between any client and any server.

ALG for IKE and ESP traffic has been created and NAT has been enhanced to implement the following:

- To enable the devices to pass IKE and ESP traffic with a source NAT pool
- To allow the device to be configured to return the same NAT-generated IP address for the same IP address without NAT ("address-persistent NAT"). As a result, the device is able to associate a client's outgoing IKE traffic with its return traffic from the server, especially when the IKE session times out and needs to be reestablished.

- The resulting ESP traffic between the client and the server is also allowed, especially in the direction from the server to the client.
- The return ESP traffic matches the following:
 - The server IP address as source IP
 - The client IP address as destination IP

Understanding IKE and ESP ALG Operation

Application Layer Gateway (ALG) for Internet Key Exchange (IKE) and Encapsulating Security Payload (ESP) traffic has the following behavior:

- An IKE and ESP ALG monitors IKE traffic between the client and the server, and it permits only one IKE Phase 2 message exchange between the client and the server at any given time.
- For a Phase 2 message:
 - If a Phase 2 message exchange between the client and server does not happen, the IKE ALG gates are opened for the relevant ESP traffic from the client to the server and from the server to the client.
 - If both IKE ALG gates are not opened successfully, or if the Phase 2 message exchange already took place, then the Phase 2 message is dropped.
- When ESP traffic hits the IKE ALG gates, sessions are created to capture subsequent ESP traffic, and to perform the proper NATing (that is, the source IP address translation from the client to the server traffic and the destination IP address translation from the server to the client traffic).
- When the ESP traffic does not hit either one or both of the gates, then the gates naturally time out.
- Once the IKE ALG gates are collapsed or timed out, another IKE Phase 2 message exchange is permitted.
- IKE *NAT-T* traffic on floating port 4500 is not processed in an IKE ALG. To support a mixture of NAT-T-capable and non-capable clients, you need to enable source NAT address persistent.

Example: Configuring the IKE and ESP ALG

IN THIS SECTION

- Requirements | 40
- Overview | 40
- Configuration | 41
- Verification | 47

This example shows how to configure the IKE and ESP ALG to pass through IKE and ESP traffic with a source NAT pool on Juniper Networks devices.

Requirements

Before you begin:

- Configure proxy ARP for all IP addresses in the source NAT pool.
- Understand the concepts behind IKE and ESP ALG. See [Understanding IKE and ESP ALG Operation](#).

Overview

IN THIS SECTION

- Topology | 41

In this example, the ALG for IKE and ESP is configured to monitor and allow IKE and ESP traffic to be exchanged between the clients and the server located on opposite sides of a Juniper Networks device.

This example shows how to configure a source NAT pool and rule set, configure a custom application to support the IKE and ESP ALG, and associate this ALG to a policy.

If you want to support a mixture of NAT-traversal (NAT-T) capable clients and noncapable clients, you must enable persistent source NAT translation (so that once a particular source NAT is associated with a given IP address, subsequent source NAT translations use the same IP address). You also must configure a custom IKE NAT traversal application to support the encapsulation of IKE and ESP in UDP port 4500. This configuration enables IKE and ESP to pass through the NAT-enabled device.

Topology

Configuration

IN THIS SECTION

- [Configuring a NAT Source Pool and Rule Set | 41](#)
- [Configuring a Custom Application and Associating it to a Policy | 43](#)
- [Configuring IKE and ESP ALG Support for Both NAT-T Capable and Noncapable Clients | 45](#)

Configuring a NAT Source Pool and Rule Set

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security nat source pool pool1 address 10.10.10.1/32 to 10.10.10.10/32
set security zones security-zone green address-book address sa1 1.1.1.0/24
set security zones security-zone red address-book address da1 2.2.2.0/24
set security nat source rule-set rs1 from zone green
set security nat source rule-set rs1 to zone red
set security nat source rule-set rs1 rule r1 match source-address 1.1.1.0/24
set security nat source rule-set rs1 rule r1 match destination-address 2.2.2.0/24
set security nat source rule-set rs1 rule r1 then source-nat pool pool1
```

Step-by-Step Procedure

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

To configure a source NAT pool:

1. Create a NAT source pool.

```
[edit ]
user@host# set security nat source pool pool1 address 10.10.10.1/32 to 10.10.10.10/32
```

2. Configure security zone address book entries.

```
[edit]
user@host# set security zones security-zone green address-book address sa1 1.1.1.0/24
user@host# set security zones security-zone red address-book address da1 2.2.2.0/24
```

3. Create a NAT source rule set.

```
[edit security nat source rule-set rs1]
user@host# set from zone green
user@host# set to zone red
user@host# set rule r1 match source-address 1.1.1.0/24
user@host# set rule r1 match destination-address 2.2.2.0/24
user@host# set rule r1 then source-nat pool pool1
```

Results

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

```
user@host# show security nat
source {
  pool pool1 {
    address {
      10.10.10.1/32 to 10.10.10.10/32;
    }
  }
}
rule-set rs1 {
  from zone green;
  to zone red;
  rule r1 {
    match {
      source-address 1.1.1.0/24;
```

```

        destination-address 2.2.2.0/24;
    }
    then {
        source-nat {
            pool {
                pool1;
            }
        }
    }
}
}
}

```

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

Configuring a Custom Application and Associating it to a Policy

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```

set applications application custom-ike-alg source-port 500 destination-port 500 protocol udp
application-protocol ike-esp-nat
set security policies from-zone green to-zone red policy pol1 match destination-address da1
set security policies from-zone green to-zone red policy pol1 match application custom-ike-alg
set security policies from-zone green to-zone red policy pol1 then permit

```

Step-by-Step Procedure

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

To configure a custom application and associate it to a policy:

1. Configure a custom application.

```
[edit]
user@host# set applications application custom-ike-alg source-port 500 destination-port 500
protocol udp application-protocol ike-esp-nat
```

2. Associate the custom application to a policy.

```
[edit security policies from-zone green to-zone red policy pol1]
user@host# set match source-address sa1
user@host# set match destination-address da1
user@host# set match application custom-ike-alg
user@host# set then permit
```

Results

From configuration mode, confirm your configuration by entering the `show applications` and `show security zones` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show applications
application custom-ike-alg {
    application-protocol ike-esp-nat;
    protocol udp;
    source-port 500;
    destination-port 500;
}
```

```
[edit]
user@host# show security zones
security-zone Trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
}
```

```

    }
  }
  interfaces {
    ge-0/0/1.0;
  }
}
security-zone green {
  address-book {
    address sa1 1.1.1.0/24;
  }
}
security-zone red {
  address-book {
    address da1 2.2.2.0/24;
  }
}
}

```

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

Configuring IKE and ESP ALG Support for Both NAT-T Capable and Noncapable Clients

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```

set security nat source address-persistent
set applications application custom-ike-natt protocol udp source-port 4500 destination-port 4500
set security policies from-zone green to-zone red policy pol1 match source-address sa1
set security policies from-zone green to-zone red policy pol1 match destination-address da1
set security policies from-zone green to-zone red policy pol1 match application custom-ike-natt
set security policies from-zone green to-zone red policy pol1 then permit

```

Step-by-Step Procedure

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

To configure IKE and ESP ALG support for both NAT-T capable and noncapable clients:

1. Globally enable persistent source NAT translation.

```
[edit]
user@host# set security nat source address-persistent
```

2. Configure the IKE NAT-T application.

```
[edit]
user@host# set applications application custom-ike-natt protocol udp source-port 4500
destination-port 4500
```

3. Associate the NAT-T application using a policy.

```
[edit security policies from-zone green to-zone red policy pol1]
user@host# set match source-address sa1
user@host# set match destination-address da1
user@host# set match application custom-ike-natt
user@host# set then permit
```

Results

From configuration mode, confirm your configuration by entering the `show security nat` and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
    address-persistent;
}
```

```
[edit]
user@host# show security policies
from-zone green to-zone red {
    policy pol1 {
        match {
            source-address sa1;
```

```

        destination-address da1;
        application [ custom-ike-alg custom-ike-natt ];
    }
    then {
        permit;
    }
}
}
default-policy {
    permit-all;
}

```

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

Verification

IN THIS SECTION

- [Verifying IKE and ESP ALG Custom Applications | 47](#)
- [Verifying the Security Policies of ALG | 48](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying IKE and ESP ALG Custom Applications

Purpose

Verify that the custom applications to support the IKE and ESP ALG are enabled.

Action

From operational mode, enter the `show security alg status` command.

```
user@host> show security alg status
```

```

ALG Status :
  DNS      : Enabled

```

```

FTP      : Enabled
H323     : Enabled
MGCP     : Enabled
MSRPC    : Enabled
PPTP     : Enabled
RSH      : Disabled
RTSP     : Enabled
SCCP     : Enabled
SIP      : Enabled
SQL      : Enabled
SUNRPC   : Enabled
TALK     : Enabled
TFTP     : Enabled
IKE-ESP  : Enabled

```

Meaning

The output shows the ALG status as follows:

- Enabled—Shows the ALG is enabled.
- Disabled—Shows the ALG is disabled.

Verifying the Security Policies of ALG

Purpose

Verify that the application custom IKE ALG and application custom IKE NATT are set.

Action

From operational mode, enter the `show security policies` command.

```
user@host> show security policies
```

```

Default policy: permit-all
From zone: green, To zone: red
Policy: pol1, State: enabled, Index: 7, Scope Policy: 0, Sequence number: 1
Source addresses: sa1
Destination addresses: da1

```

```
Applications: custom-ike-alg, custom-ike-natt
Action: permit
```

Meaning

The sample output shows that custom IKE ALG and custom IKE NATT applications are set.

Example: Enabling the IKE and ESP ALG and Setting Timeouts

IN THIS SECTION

- Requirements | 49
- Overview | 49
- Configuration | 50
- Verification | 51

This example shows how to enable the IKE and ESP ALG and set the timeout values to allow time for the ALG to process ALG state information, ESP gates, and ESP sessions.

Requirements

Understand the concepts behind ALG for IKE and ESP. See [Understanding IKE and ESP ALG Operation](#).

Overview

The IKE and ESP ALG processes all traffic specified in any policy to which the ALG is attached. In this example, you configure the **set security alg ike-esp-nat enable** statement so the current default IPsec pass-through behavior is disabled for all IPsec pass-through traffic, regardless of policy.

You then set the timeout values to allow time for the IKE and ESP ALG to process ALG state information, ESP gates, and ESP sessions. In this example, you set the timeout of ALG state information. The timeout range is 180 through 86400 seconds. The default timeout is 14400 seconds. You then set the timeout of the ESP gates created after an IKE Phase 2 exchange has completed. The timeout range is 2 through 30 seconds. The default timeout is 5 seconds. Finally, you set the idle timeout of the ESP sessions created from the IPsec gates. If no traffic hits the session, it is aged out after this period of time. The timeout range is 60 through 2400 seconds. The default timeout is 1800 seconds.

Configuration

IN THIS SECTION

- [Procedure](#) | 50

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security alg ike-esp-nat enable
set security alg ike-esp-nat esp-gate-timeout 20
set security alg ike-esp-nat esp-session-timeout 2400
set security alg ike-esp-nat state-timeout 360
```

Step-by-Step Procedure

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

To enable the IKE and ESP ALG and set the timeout values:

1. Enable the IKE and ESP ALG.

```
[edit]
user@host# set security alg ike-esp-nat enable
```

2. Set the timeout for the ALG state information.

```
[edit security alg ike-esp-nat]
user@host# set state-timeout 360
```

3. Set the timeout for the ESP gates created after an IKE Phase 2 exchange has completed.

```
[edit security alg ike-esp-nat]
user@host# set esp-gate-timeout 20
```

4. Set the idle timeout for the ESP sessions created from the IPsec gates.

```
[edit security alg ike-esp-nat]
user@host# set esp-session-timeout 2400
```

Results

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

```
[edit]
user@host# show security alg
ike-esp-nat {
    enable;
    state-timeout 360;
    esp-gate-timeout 20;
    esp-session-timeout 2400;
}
```

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

Verification

IN THIS SECTION

- [Verifying the ALG for IKE and ESP and Timeout Settings | 52](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying the ALG for IKE and ESP and Timeout Settings

Purpose

Verify that the ALG for IKE and ESP is enabled and the timeout settings for this feature are correct.

Action

From operational mode, enter the `show security alg ike-esp-nat` command.

SEE ALSO

| [Introduction to NAT](#)

Platform-Specific IKE ALG Behavior

Use [Feature Explorer](#) to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platform:

Platform	Difference
SRX Series	<ul style="list-style-type: none">SRX1500, SRX5600, and SRX5800 devices that support IKE, supports IKE negotiations with NAT traversal. These negotiations fail if the IKE peer is behind a NAT device that changes the source IP address during negotiation. For example, a NAT device with DIP changes the source IP because the IKE protocol switches the UDP port from 500 to 4500.

RELATED DOCUMENTATION

| [RPC ALG | 72](#)
| [RSH ALG | 87](#)

PPTP ALG

IN THIS SECTION

- [Understanding the PPTP ALG | 53](#)
- [Example: Configuring the PPTP ALG | 54](#)

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.

Understanding the PPTP ALG

IN THIS SECTION

- [Understanding IPv6 Support for the PPTP ALG | 53](#)

The Point-to-Point Tunneling Protocol (PPTP) ALG is used for tunneling Point-to-Point Protocol (PPP) packets over an IP network. The PPTP ALG is often used to implement a client/server architecture, a PPTP network server, and a PPTP access concentrator.

The PPTP ALG processes PPTP packets, performs Network Address Translation (NAT), open pinholes for new data connections between a client and a server, and transfers data between a client and a server located on opposite sides of a Juniper Networks device.

Understanding IPv6 Support for the PPTP ALG

The PPTP ALG uses TCP port 1723 to connect and disconnect a client and a server. The PPTP ALG supports IPv6 data packets.

The PPTP ALG with IPv6 support, parses both IPv4 and IPv6 PPTP packets, performs NAT, and then opens a pinhole for the data tunnel.

The PPTP ALG with IPv6 support does not support NAT-PT and NAT64, because PPP packets are compressed with Microsoft Point-to-Point Encryption (MPPE) protocol after the tunnel is set up; therefore translation of the IP header in the PPP package cannot be handled.

- The PPTP ALG with IPv6 support has the following limitation:
 - Because PPP packets are compressed with Microsoft Point-to-Point Encryption (MPPE) protocol after the tunnel is set up, translation of the IP header in the PPP package cannot be handled; therefore, to make sure PPTP connection works well, the PPTP client must be able to work in dual stack mode. So that an IPv6 PPTP client can accept an IPv4 address for PPP tunnel interface, by which it can communicate with the IPv4 PPTP server without IP address translation for PPP packets.

The flow module supports IPv6 to parse the GRE packet and use the GRE call ID as fake port information to search the session table and gate table.



NOTE: The PPTP ALG can support NAT64 in a specific scenario in which translation of the IP header in the PPP package is not required—that is, if the PPTP client works in dual-stack mode in the IPv6 network and server in the IPv4 network.

Example: Configuring the PPTP ALG

IN THIS SECTION

- [Requirements | 55](#)
- [Overview | 55](#)
- [Configuration | 57](#)
- [Verification | 68](#)

The PPTP ALG processes PPTP packets, performs NAT, and open pinholes for new data connections between a client and a server.

This example shows how to configure the PPTP ALG in route or NAT mode. The configuration allows PPTP traffic to pass through a device, transferring data between a client and a server located on opposite sides of a Juniper Networks device.

Requirements

This example uses the following hardware and software components:

- An SRX Series Firewall
- Two PCs (client and server)

Before you begin:

- Understand the concepts behind ALGs. See ["ALG Overview" on page 2](#).
- Understand the basics of PPTP ALG. See ["Understanding the PPTP ALG" on page 53](#).

Overview

IN THIS SECTION

- [Topology | 55](#)

In this example, first you configure network interfaces on the device, create security zones and assign interfaces to the zones, and configure a policy to allow PPTP traffic to go through an SRX Series Firewall.

Then you create a static NAT rule set rs1 with a rule r1 to match with the destination address 30.5.2.120/32, and you create a static NAT prefix with address 10.5.1.120/32.

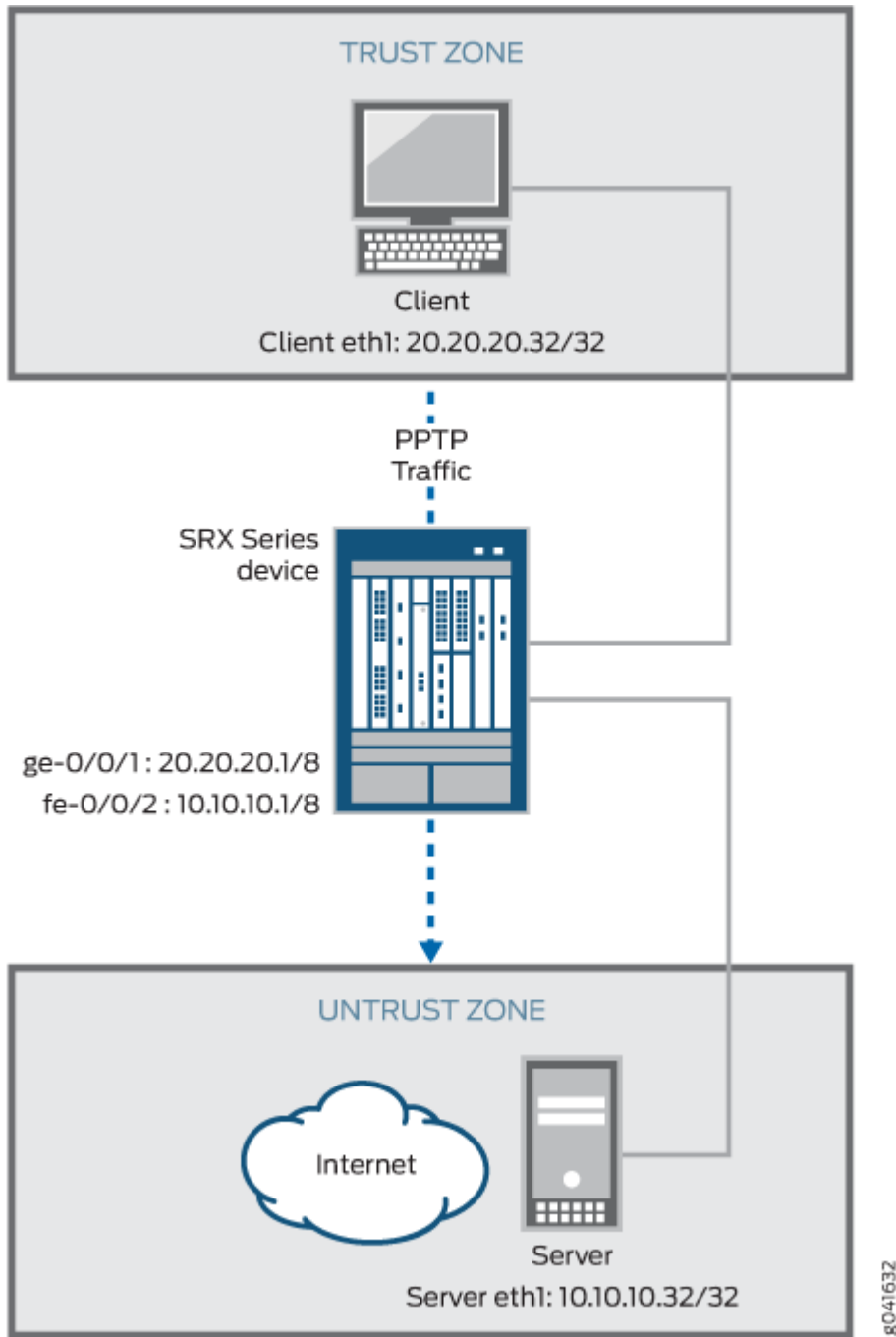
Next you create a source NAT pool src-p1 with a source rule set src-rs1 to translate packets from zone trust to zone untrust. For matching packets, the source address is translated to an IP address in the src-p1 pool.

Then you create a destination NAT pool des-p1 with a destination rule set des-rs1 to translate packets from zone trust to destination address 30.5.1.120/32. For matching packets, the destination address is translated to an IP address in the des-p1 pool. Finally, you configure PPTP ALG trace options.

Topology

[Figure 4 on page 56](#) shows the PPTP ALG topology.

Figure 4: PPTP ALG Topology



Configuration

IN THIS SECTION

- [Configuring a Route Mode | 57](#)
- [Configuring a Static NAT Rule Set | 60](#)
- [Configuring a Source NAT Pool and Rule Set | 62](#)
- [Configuring a Destination NAT Pool and Rule Set | 64](#)
- [Configuring PPTP ALG trace options | 66](#)

To configure the PPTP ALG, perform these tasks:

Configuring a Route Mode

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 20.20.20.1/8
set interfaces fe-0/0/2 unit 0 family inet address 10.10.10.1/8
set security zones security-zone trust interfaces ge-0/0/1 host-inbound-traffic system-services
all
set security zones security-zone trust interfaces ge-0/0/1 host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-0/0/2 host-inbound-traffic system-
services all
set security zones security-zone untrust interfaces fe-0/0/2 host-inbound-traffic protocols all
set security policies from-zone trust to-zone untrust policy pptp match source-address any
set security policies from-zone trust to-zone untrust policy pptp match destination-address any
set security policies from-zone trust to-zone untrust policy pptp match application junos-pptp
set security policies from-zone trust to-zone untrust policy pptp then permit
```

Step-by-Step Procedure

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

To configure route mode:

1. Configure interfaces.

```
[edit interfaces]
user@host#set ge-0/0/1 unit 0 family inet address 20.20.20.1/8
user@host#set fe-0/0/2 unit 0 family inet address 10.10.10.1/8
```

2. Configure zones and assign interfaces to the zones.

```
[edit security zones security-zone trust]
user@host#set interfaces ge-0/0/1 host-inbound-traffic system-services all
user@host#set interfaces ge-0/0/1 host-inbound-traffic protocols all
[edit security zones security-zone untrust]
user@host#set interfaces fe-0/0/2 host-inbound-traffic system-services all
user@host#set interfaces fe-0/0/2 host-inbound-traffic protocols all
```

3. Configure a PPTP policy that allows PPTP traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host#set policy pptp match source-address any
user@host#set policy pptp match destination-address any
user@host#set policy pptp match application junos-pptp
user@host#set policy pptp then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

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

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

```

        family inet {
            address 20.20.20.1/8;
        }
    }
}
fe-0/0/2 {
    unit 0 {
        family inet {
            address 10.10.10.1/8;
        }
    }
}
...

```

```

[edit]
user@host# show security zones
security-zone trust {
    ....
    interfaces {
        ge-0/0/1 {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
        }
    }
}
...
security-zone untrust {
    interfaces {
        fe-0/0/2 {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
        }
    }
}

```



```

    }
  }
}

```

```

[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy pptp {
    match {
      source-address any;
      destination-address any;
      application junos-pptp;
    }
    then {
      permit;
    }
  }
}

```

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

Configuring a Static NAT Rule Set

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```

set security nat static rule-set rs1 from zone trust
set security nat static rule-set rs1 rule r1 match destination-address 30.5.2.120/32
set security nat static rule-set rs1 rule r1 then static-nat prefix 10.5.1.120/32

```

Step-by-Step Procedure

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

To configure a static NAT rule set:

1. Create a static NAT rule set.

```
[edit security nat static rule-set rs1]
user@host# set from zone trust
```

2. Define the rule to match with the destination address.

```
[edit security nat static rule-set rs1]
user@host# set rule r1 match destination-address 30.5.2.120/32
```

3. Define the static NAT prefix for the device.

```
[edit security nat static rule-set rs1]
user@host# set rule r1 then static-nat prefix 10.5.1.120/32
```

Results

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

```
[edit]
user@host# show security nat
static {
  rule-set rs1 {
    from zone trust;
    rule r1 {
      match {
        destination-address 30.5.2.120/32;
      }
      then {
        static-nat {
```


2. Create a source NAT rule set.

```
[edit security nat source ]
user@host# set rule-set src-rs1 from zone trust
user@host# set rule-set src-rs1 to zone untrust
```

3. Configure a rule that matches packets and translates the source address to an address in the source pool.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule src-r1 match source-address 20.5.1.120/32
```

4. Configure a rule that matches packets and translates the destination address to an address in the source pool.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule src-r1 match destination-address 10.5.2.120/32
```

5. Configure a source NAT pool in the rule.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule src-r1 then source-nat pool src-p1
```

Results

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

```
[edit]
user@host# show security nat
source {
  pool src-p1 {
    address {
      30.5.1.120/32;
    }
  }
}
rule-set src-rs1 {
```

```

from zone trust;
to zone untrust;
rule src-r1 {
    match {
        source-address 20.5.1.120/32;
        destination-address 10.5.2.120/32;
    }
    then {
        source-nat {
            pool {
                src-p1;
            }
        }
    }
}
}
}

```

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

Configuring a Destination NAT Pool and Rule Set

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```

set security nat destination pool des-p1 address 10.5.1.120/32
set security nat destination rule-set des-rs1 from zone trust
set security nat destination rule-set des-rs1 rule des-r1 match source-address 20.5.1.120/32
set security nat destination rule-set des-rs1 rule des-r1 match destination-address 30.5.1.120/32
set security nat destination rule-set des-rs1 rule des-r1 then destination-nat pool des-p1

```

Step-by-Step Procedure

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

To configure a destination NAT pool and rule set:

1. Create a destination NAT pool.

```
[edit security nat destination]
user@host# set pool des-p1 address 10.5.1.120/32
```

2. Create a destination NAT rule set.

```
[edit security nat destination]
user@host# set rule-set des-rs1 from zone trust
```

3. Configure a rule that matches packets and translates the source address to the address in the pool.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 match source-address 20.5.1.120/32
```

4. Configure a rule that matches packets and translates the destination address to the address in the pool.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 match destination-address 30.5.1.120/32
```

5. Configure a source NAT pool in the rule.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 then destination-nat pool des-p1
```

Results

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

```
[edit]
user@host# show security nat
destination {
  pool des-p1 {
    address {
```

```

        10.5.1.120/32;
    }
}
rule-set des-rs1 {
    from zone trust;
    rule des-r1 {
        match {
            source-address 20.5.1.120/32;
            destination-address 30.5.1.120/32;
        }
        then {
            destination-nat {
                pool {
                    des-p1;
                }
            }
        }
    }
}
}
}
}
}

```

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

Configuring PPTP ALG trace options

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```

set security alg pptp traceoptions flag all
set security alg traceoptions file trace
set security alg traceoptions file size 1g
set security alg traceoptions level verbose

```

Step-by-Step Procedure

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

To configure PPTP ALG trace options:

1. Enable PPTP ALG trace options.

```
[edit security alg]
user@host#set pptp traceoptions flag all
```

2. Configure a filename to receive output from the tracing operation.

```
[edit security alg]
user@host#set traceoptions file trace
```

3. Specify the maximum trace file size.

```
[edit security alg]
user@host#set traceoptions file size 1g
```

4. Specify the level of tracing output.

```
[edit security alg]
user@host#set traceoptions level verbose
```

Results

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

```
[edit]
user@host# show security alg
traceoptions {
  file trace size 1g;
  level verbose;
```



```
}
pptp traceoptions flag all;
```

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

Verification

IN THIS SECTION

- [Verifying the PPTP ALG Control Session | 68](#)
- [Verifying the PPTP ALG Flow Gate Information | 69](#)
- [Verifying PPTP ALG | 70](#)
- [Verifying the PPTP Resource Manager Group | 71](#)
- [Verifying the PPTP Resource Information | 71](#)

Confirm that the configuration is working properly.

Verifying the PPTP ALG Control Session

Purpose

Verify that the PPTP control session is created and all the PPTP control and data sessions are created.

Action

From operational mode, enter the `show security flow session` command.

```
user@host>show security flow session
SSession ID: 57, Policy name: pptp, Timeout: 1787
Resource information : PPTP ALG, 1, 0
In: 20.20.20.32/3905 --> 10.10.10.32/1723;tcp, If: ge-0/0/1.0 Pkts: 6, Bytes: 584
Out: 10.10.10.32/1723 --> 20.20.20.32/3905;tcp, If: fe-0/0/2.0 Pkts: 4, Bytes: 352

Session ID: 58, Policy name: pptp, Timeout: 1799
In: 20.20.20.32/0 --> 10.10.10.32/256;gre, If: ge-0/0/1.0
Out: 10.10.10.32/256 --> 20.20.20.32/65001;gre, If: fe-0/0/2.0
```

```
Session ID: 59, Policy name: pptp, Timeout: 1787
In: .10.10.10.32/0 --> 20.20.20.32/260;gre, If: ge-0/0/1.0
Out: 20.20.20.32/260 --> 10.10.10.32/65000;gre, If: fe-0/0/2.0
```

Meaning

- **Session ID**—Number that identifies the session. Use this ID to get more information about the session such as policy name or number of packets in and out.
- **Policy name**—Policy name that permitted the traffic.
- **In**—Incoming flow (source and destination IP addresses with their respective source and destination port numbers, session is TCP, and the source interface for this session is ge-0/0/1.0).
- **Out**—Reverse flow (source and destination IP addresses with their respective source and destination port numbers, session is TCP, and destination interface for this session is fe-0/0/2.0).

Verifying the PPTP ALG Flow Gate Information

Purpose

Verify that the flow gate is opened for TCP data channel connection.

Action

From operational mode, enter the `show security flow gate` command.

```
user@host>show security flow gate

Hole: 20.0.172.24-20.0.172.24/0-0->21.0.172.38-21.0.172.38/25750-25750
Translated: 2015::172:24/65000->2005::172:108/360
Protocol: gre
Application: PPTP ALG/69
Age: 118 seconds
Flags: 0x0080
Zone: trust
Reference count: 1
Resource: 12-1-1

Hole: 2005::172:108-0-0->2015::172:24-2432-2432
Translated: 21.0.172.38/65001->20.0.172.24/2432
Protocol: gre
```

```

Application: PPTP ALG/69
Age: 120 seconds
Flags: 0x8080
Zone: untrust
Reference count: 1
Resource: 12-1-2

```

```

Valid gates: 2
Pending gates: 0
Invalidated gates: 0
Gates in other states: 0
Total gates: 2

```

Verifying PPTP ALG

Purpose

Verify that the PPTP ALG is enabled.

Action

From operational mode, enter the `show security alg status` command.

```

user@host>show security alg status
ALG Status :
  PPTP      : Enabled
  RSH       : Disabled
  RTSP      : Enabled
  SCCP      : Enabled
  SIP       : Enabled
  TALK      : Enabled
  TFTP      : Enabled
  IKE-ESP   : Disabled

```

Meaning

The output shows the PPTP ALG status as follows:

- Enabled—Shows the PPTP ALG is enabled.
- Disabled—Shows the PPTP ALG is disabled.

Verifying the PPTP Resource Manager Group

Purpose

Verify the total number of resource manager groups and active groups that are used by the PPTP ALG.

Action

From operational mode, enter the `show security resource-manager group active` command.

```
user@host>show security resource-manager group active
Group ID 1: Application - PPTP ALG
Total groups 19763, active groups 1
```

Verifying the PPTP Resource Information

Purpose

Verify the total number of resources and active resources that are used by the PPTP ALG.

Action

From operational mode, enter the `show security resource-manager resource active` command.

```
user@host>show security resource-manager resource active
Resource ID 2: Group ID - 1, Application - PPTP ALG

Resource ID 1: Group ID - 1, Application - PPTP ALG
Total Resources 93286, active resources 2
```

RELATED DOCUMENTATION

[RSH ALG | 87](#)

[RTSP ALG | 106](#)

RPC ALG

IN THIS SECTION

- [Understanding RPC ALGs | 72](#)
- [Understanding Sun RPC ALGs | 73](#)
- [Enabling Sun RPC ALGs | 74](#)
- [Customizing Sun RPC Applications \(CLI Procedure\) | 74](#)
- [Understanding Sun RPC Services | 75](#)
- [Understanding Microsoft RPC ALGs | 78](#)
- [Enabling Microsoft RPC ALGs | 79](#)
- [Configuring the Microsoft RPC ALG | 80](#)
- [Understanding Microsoft RPC Services | 83](#)
- [Customizing Microsoft RPC Applications \(CLI Procedure\) | 85](#)

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 the allowed program numbers.

Understanding RPC ALGs

Junos OS supports basic Remote Procedure Call Application Layer Gateway (RPC ALG) services. RPC is a protocol that allows an application running in one address space to access the resources of applications running in another address space as if the resources were local to the first address space. The RPC ALG is responsible for RPC packet processing.

The RPC ALG in Junos OS supports the following services and features:

- Sun Microsystems RPC Open Network Computing (ONC)
- Microsoft RPC Distributed Computing Environment (DCE)
- Dynamic port negotiation

- Ability to allow and deny specific RPC services
- Static Network Address Translation (NAT) and source NAT (with no port translation)
- RPC applications in security policies

Use the RPC ALG if you need to run RPC-based applications such as NFS or Microsoft Outlook. The RPC ALG functionality is enabled by default.

Understanding Sun RPC ALGs

Sun Microsystems Remote Procedure Call (Sun RPC)—also known as Open Network Computing Remote Procedure Call (ONC RPC)—provides a way for a program running on one host to call procedures in a program running on another host. Because of the large number of RPC services and the need to broadcast, the transport address of an RPC service is dynamically negotiated based on the service's program number and version number. Several binding protocols are defined for mapping the RPC program number and version number to a transport address.

Junos OS supports the Sun RPC as a predefined service and allows and denies traffic based on a security policy you configure. The Application Layer Gateway (ALG) provides the functionality for Juniper Networks devices to handle the dynamic transport address negotiation mechanism of the Sun RPC and to ensure program number-based security policy enforcement. You can define a security policy to permit or deny all RPC requests, or to permit or deny by specific program number. The ALG also supports route mode and Network Address Translation (NAT) mode for incoming and outgoing requests.

When an application or a PC client calls a remote service, it needs to find the transport address of the service. In the case of TCP/UDP, the address is a port number. A typical procedure for this case is as follows:

1. The client sends the GETPORT message to the RPCBIND service on the remote machine. The GETPORT message contains the program number, and version and procedure number of the remote service it is attempting to call.
2. The RPCBIND service replies with a port number.
3. The client calls the remote service using the port number returned.
4. The remote service replies to the client.

A client also can use the CALLIT message to call the remote service directly, without determining the port number of the service. In this case, the procedure is as follows:

1. The client sends a CALLIT message to the RPCBIND service on the remote machine. The CALLIT message contains the program number and the version and procedure number of the remote service it attempting to call.

2. RPCBIND calls the service for the client.
3. RPCBIND replies to the client if the call has been successful. The reply contains the call result and the service's port number.

The Sun RPC ALG dynamically allocates new mapping entries instead of using a default size (512 entries). It also offers a flexible time-based RPC mapping entry that removes the mapping entry (auto-clean) without affecting the associated active RPC sessions, including both control session and data session.

Starting in Junos OS 15.1X49-D10 and Junos OS Release 17.3R1, you can define the Sun RPC mapping entry ageout value. Use the `set security alg sunrpc map-entry-timeout value` command. The ageout value ranges from 1 hour to 72 hours, and the default value is 32 hours. If the Sun RPC ALG service does not trigger the control negotiation even after 72 hours, the maximum RPC ALG mapping entry value times out and the new data connection to the service fails.

Enabling Sun RPC ALGs

The Sun RPC ALG is enabled by default and requires no configuration.

Enabling Sun RPC ALGs (CLI Procedure)

To disable the Sun RPC ALG, enter the following command:

```
user@host# set security alg sunrpc disable
```

To re-enable the Sun RPC ALG, enter the following command:

```
user@host# delete security alg sunrpc
```

Customizing Sun RPC Applications (CLI Procedure)

All Sun RPC applications can be customized by using a predefined application set.

For example, an application can be customized to open the control session only and not allow any data sessions:

```
application-set junos-sun-rpc {
    application junos-sun-rpc-tcp;
    application junos-sun-rpc-udp;
}
```

In the following example, the predefined application set allows data sessions only. It will not work without the control session:

```
application-set junos-sun-rpc-portmap {
    application junos-sun-rpc-portmap-tcp;
    application junos-sun-rpc-portmap-udp;
}
```

To customize all Sun RPC applications with predefined application sets, use both application sets in the policy:

```
application-set [junos-sun-rpc junos-sun-rpc-portmap]
```



NOTE: MS RPC applications are customized in the same way as Sun RPC applications.

Understanding Sun RPC Services

Sun RPC, also known as Open Network computing remote procedure call (ONC RPC), provides a way for a program running on one host to call procedures in a program running on another host. Sun RPC services are defined by a program identifier. The program identifier is independent of any transport address, and most of the Sun RPC sessions are initiated through TCP or UDP port 111. Each host links the required RPC service to a dynamic TCP or UDP port that is negotiated over the port 111 control channel, allowing the client to connect to either TCP or UDP port 111.

Predefined Sun Microsystems remote procedure call (Sun RPC) services include:

- junos-sun-rpc-tcp
- junos-sun-rpc-udp

The Sun RPC ALG can be applied by using the following methods:

- ALG default application—Use one of the following predefined applications for control and data connections in your policy:
 - `junos-sun-rpc-any-tcp`
 - `junos-sun-rpc-any-udp`
 - `junos-sun-rpc-mountd-tcp`
 - `junos-sun-rpc-mountd-udp`
 - `junos-sun-rpc-nfs-tcp`
 - `junos-sun-rpc-nfs-udp`
 - `junos-sun-rpc-nlockmgr-tcp`
 - `junos-sun-rpc-nlockmgr-udp`
 - `junos-sun-rpc-portmap-tcp`
 - `junos-sun-rpc-portmap-udp`
 - `junos-sun-rpc-rquotad-tcp`
 - `junos-sun-rpc-rquotad-udp`
 - `junos-sun-rpc-ruserd-tcp`
 - `junos-sun-rpc-ruserd-udp`
 - `junos-sun-rpc-sadmind-tcp`
 - `junos-sun-rpc-sadmind-udp`
 - `junos-sun-rpc-sprayd-tcp`
 - `junos-sun-rpc-sprayd-udp`
 - `junos-sun-rpc-status-tcp`
 - `junos-sun-rpc-status-udp`
 - `junos-sun-rpc-walld-tcp`
 - `junos-sun-rpc-walld-udp`
 - `junos-sun-rpc-ybind-tcp`

- junos-sun-rpc-ypbind-udp
- junos-sun-rpc-ypserv-tcp
- junos-sun-rpc-ypserv-udp
- Default control application—Use the predefined control through junos-sun-rpc:
 - Create an application for data (USER_DEFINED_DATA). You can make a set of your own data (for example, my_rpc_application_set) and use it in the policy.
 - ALG default application set—Use the predefined application set for control and customized data application in the policy:
 - junos-sun-rpc (for control sessions)
 - junos-sun-rpc-any
 - junos-sun-rpc-mountd
 - junos-sun-rpc-nfs
 - junos-sun-rpc-nfs-access
 - junos-sun-rpc-nlockmgr
 - junos-sun-rpc-portmap (for data sessions)
 - junos-sun-rpc-rquotad
 - junos-sun-rpc-ruserd
 - junos-sun-rpc-sadmind
 - junos-sun-rpc-sprayd
 - junos-sun-rpc-status
 - junos-sun-rpc-walld
 - junos-sun-rpc-ypbind
 - junos-sun-rpc-ypserv
- Custom control and custom data application—Use a customized application:
 - Create an application for control (USER_DEFINED_CONTROL) and data (USER_DEFINED_DATA).
 - In the policy, use the user-defined application set for a control and customized data application:
 - USER_DEFINED_CONTROL

- USER_DEFINED_DATA

Table 2 on page 78 lists predefined Sun RPC services, a program identifier associated with each service, and a description of each service.

Table 2: Predefined Sun RPC Services

Service	Program ID	Description
PORTMAP	100000	Sun RPC Portmapper protocol is a TCP or UDP port-based service that includes TCP or UDP port 111.
NFS	100003	Sun RPC Network File System.
MOUNT	100005	Sun RPC mount process.
YPBIND	100007	Sun RPC Yellow Page Bind service.
STATUS	100024	Sun RPC status.

Understanding Microsoft RPC ALGs

Microsoft Remote Procedure Call (MS-RPC) is the Microsoft implementation of the Distributed Computing Environment (DCE) RPC. Like the Sun RPC, MS-RPC provides a way for a program running on one host to call procedures in a program running on another host. Because of the large number of RPC services and the need to broadcast, the transport address of an RPC service is dynamically negotiated based on the service program's universal unique identifier (UUID). The specific UUID is mapped to a transport address.

Junos OS devices running Junos OS support MS-RPC as a predefined service and allow and deny traffic based on a policy you configure. The Application Layer Gateway (ALG) provides the functionality for Juniper Networks devices to handle the dynamic transport address negotiation mechanism of the MS-RPC, and to ensure UUID-based security policy enforcement. You can define a security policy to permit or deny all RPC requests, or to permit or deny by specific UUID number. The ALG also supports route mode and Network Address Translation (NAT) mode for incoming and outgoing requests.

When both the MS-RPC client and MS-RPC server are 64 bit capable (such as MS Exchange 2008), they negotiate to use NDR64 transfer syntax during the network communication. when you use NDR64, the

interface parameters should be encoded according to NDR64 syntax, because the packet format for NDR64 is different from the packet format for NDR20 (32 bit version).

In MS-RPC, there is a remote activation interface of the DCOM Remote Protocol called ISystemActivator (also known as IRemoteSCMAActivator). It is used by the Windows Management Instrumentation Command-line (WMIC), Internet Information Services (IIS), and many other applications that are used extensively.

The MS-RPC ALG dynamically allocates new mapping entries instead of using a default size (512 entries). It also offers a flexible time-based RPC mapping entry that removes the mapping entry (auto-clean) without affecting the associated active RPC sessions, including both control session and data session.

Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, you can define the MS-RPC mapping entry ageout value. Use the `set security alg msrpc map-entry-timeout value` command. The ageout value ranges from 1 hour to 72 hours, and the default value is 32 hours. If the MS-RPC ALG service does not trigger the control negotiation even after 72 hours, the maximum MS-RPC ALG mapping entry value times out and the new data connection to the service fails.

Enabling Microsoft RPC ALGs

The MS-RPC ALG is enabled by default and requires no configuration.

Enabling Microsoft RPC ALGs (CLI Procedure)

To disable the Microsoft RPC ALG, enter the following command:

```
user@host# set security alg msrpc disable
```

To reenable the Microsoft RPC ALG, enter the following command:

```
user@host# delete security alg msrpc
```

Configuring the Microsoft RPC ALG

IN THIS SECTION

- [Configuring the MS-RPC ALG with a Predefined Microsoft Application | 80](#)
- [Configuring the MS-RPC ALG with a Wildcard UUID | 81](#)
- [Configuring the MS-RPC ALG with a Specific UUID | 81](#)

You can configure the Microsoft RPC ALG using the following three methods:

Configuring the MS-RPC ALG with a Predefined Microsoft Application

There are several predefined MS applications. To view the predefined Microsoft applications from the CLI, enter the `show configuration groups junos-defaults` command.

```
user@host> show security policies
  from-zone trust to-zone untrust {
    policy p1 {
      match {
        source-address any;
        destination-address any;
        application junos-ms-rpc-msexchange;
      }
      then {
        permit;
      }
    }
  }
}
```

After you commit the configuration, from the CLI, enter the `show security alg msrpc object-id-map` command to view the output.

```
user@host> show security alg msrpc object-id-map
UUID                                OID
1544f5e0-613c-11d1-93df-00c04fd7bd09  0x80000001
a4f1db00-ca47-1067-b31f-00dd010662da  0x80000002
f5cc5a18-4264-101a-8c59-08002b2f8426  0x80000003
```

The output shows that the UUID has been applied for the policy.

Configuring the MS-RPC ALG with a Wildcard UUID

To permit the configuration for any MS RPC application, add the application `junos-ms-rpc-any` statement to the Permit configuration.

```
user@host> show security policies
  from-zone trust to-zone untrust {
    policy p1 {
      match {
        source-address any;
        destination-address any;
        application junos-ms-rpc-any;
      }
      then {
        permit;
      }
    }
  }
}
```

After you commit the configuration, from the CLI, enter the `show security alg msrpc object-id-map` command to view the output.

```
user@host> show security alg msrpc object-id-map
UUID                                OID
ffffffff-ffff-ffff-ffff-fffffffffff 0x80000004
```

Configuring the MS-RPC ALG with a Specific UUID

For applications that have not been predefined, you need to manually configure a specific UUID. For example, to permit a NETLOGON application that has not been predefined, you add the application `msrpc-netlogon` statement to the Permit configuration.

In Junos OS Release 15.1X49-D90 and earlier, on all SRX Series Firewalls, the custom application universal unique identifier (UUID) of Microsoft remote procedure call (MS-RPC) with leading zeros and the nil UUID (00000000-0000-0000-0000-000000000000) might match all TCP traffic and referenced policies allowing all TCP traffic instead of entering MS-RPC ALG check.

Starting with Junos OS Release 15.1X49-D100 and Junos OS Release 17.3R1, the custom application UUID with leading zeros does not match all TCP traffic and referenced policies, which will enter MS-RPC ALG check. This new application does not allow the nil UUID.

```
user@host> show applications
  application msrpc-netlogon {
    term t1 protocol tcp uuid 12345678-1234-abcd-ef00-01234567cffb;
    term t2 protocol udp uuid 12345678-1234-abcd-ef00-01234567cffb;
    term t3 protocol tcp uuid 12345778-1234-abcd-ef00-0123456789ab;
  }
user@host> show security policies
  from-zone trust to-zone untrust {
    match {
      source-address any;
      destination-address any;
      application msrpc-netlogon;
    }
    then {
      permit;
    }
  }
}
```

After you commit the configuration, from the CLI, enter the `show security alg msrpc object-id-map` command to verify the Microsoft universal unique identifier to Object ID (UUID-to-OID) mapping table. The Microsoft RPC ALG monitors packets on TCP port 135.

```
user@host> show security alg msrpc object-id-map
UUID                                OID
12345778-1234-abcd-ef00-0123456789ab 0x80000006
12345678-1234-abcd-ef00-01234567cffb 0x80000005
be617c0-31a5-11cf-a7d8-00805f48a135 0x80000020
e3514235-4b06-11d1-ab04-00c04fc2dcd2 0x80000002
67df7c70-0f04-11ce-b13f-00aa003bac6c 0x80000014
```



NOTE: The `show security alg msrpc object-id-map` CLI command has a chassis cluster node option to permit the output to be restricted to a particular node or to query the entire cluster. The `show security alg msrpc object-id-map node` CLI command options are `<node-id | all | local | primary>`.

Understanding Microsoft RPC Services

MS-RPC is the Microsoft implementation of the Distributed Computing Environment (DCE) RPC. Like the Sun RPC, the MS-RPC provides a way for a program running on one host to call procedures in a program running on another host. The MS-RPC is dynamically negotiated based on the service program's universal unique identifier (UUID). The specific UUID is mapped to a transport address.

In Junos OS Release 15.1X49-D90 and earlier, on all SRX Series Firewalls, the custom application universal unique identifier (UUID) of Microsoft remote procedure call (MS-RPC) with leading zeros and the nil UUID (00000000-0000-0000-0000-000000000000) might match all TCP traffic and referenced policies allowing all TCP traffic instead of entering MS-RPC ALG check.

Starting with Junos OS Release 15.1X49-D100 and Junos OS Release 17.3R1, the custom application UUID with leading zeros does not match all TCP traffic and referenced policies, which will enter MS-RPC ALG check. This new application does not allow the nil UUID.

Predefined Microsoft remote procedure call (MS-RPC) services include:

- junos-ms-rpc-epm
- junos-ms-rpc-tcp
- junos-ms-rpc-udp

MS-RPC application defaults include:

- junos-ms-rpc-iis-com-1
- junos-ms-rpc-iis-com-adminbase
- junos-ms-rpc-msexchange-directory-nsp
- junos-ms-rpc-msexchange-directory-rfr
- junos-ms-rpc-msexchange-info-store
- junos-ms-rpc-uuid-any-tcp
- junos-ms-rpc-uuid-any-udp
- junos-ms-rpc-wmic-admin
- junos-ms-rpc-wmic-admin2
- junos-ms-rpc-wmic-mgmt
- junos-ms-rpc-wmic-webm-callresult

- junos-ms-rpc-wmic-webm-classobject
- junos-ms-rpc-wmic-webm-level1login
- junos-ms-rpc-wmic-webm-login-clientid
- junos-ms-rpc-wmic-webm-login-helper
- junos-ms-rpc-wmic-webm-objectsink
- junos-ms-rpc-wmic-webm-refreshing-services
- junos-ms-rpc-wmic-webm-remote-refresher
- junos-ms-rpc-wmic-webm-services
- junos-ms-rpc-wmic-webm-shutdown

MS-RPC application-set defaults include:

- junos-ms-rpc
- junos-ms-rpc-any
- junos-ms-rpc-iis-com
- junos-ms-rpc-msexchange
- junos-ms-rpc-wmic

[Table 3 on page 84](#) lists predefined MS-RPC services, UUID values associated with each service, and a description of each service.

Table 3: Predefined MS-RPC services

Service	UUID	Description
EPM	e1af8308-5d1f-11c9-91a4-08002b14a0fa	MS-RPC Endpoint Mapper (EPM) protocol is a TCP/UDP port-based service that includes TCP/UDP port 135.
EXCHANGE-DATABASE	1a190310-bb9c-11cd-90f8-00aa00466520	Microsoft Exchange Database service.

Table 3: Predefined MS-RPC services (Continued)

Service	UUID	Description
EXCHANGE-DIRECTORY	f5cc5a18-4264-101a-8c59-08002b2f8426 f5cc5a7c-4264-101a-8c59-08002b2f8426 f5cc59b4-4264-101a-8c59-08002b2f8426	Microsoft Exchange Directory service.
WIN-DNS	50abc2a4-574d-40b3-9d66-ee4fd5fba076	Microsoft Windows DNS server.
WINS	5f52c28-7f9f-101a-b52b-08002b2efabe 811109bf-a4e1-11d1-ab54-00a0c91e9b45	Microsoft WINS service.
WMIC-Webm-Level1Login	f309ad18-d86a-11d0-a075-00c04fb68820	This service allows users to connect to the management services interface in a particular namespace.

Customizing Microsoft RPC Applications (CLI Procedure)

MS-RPC applications are customized in the same way as SUN RPC applications.

MS-RPC services in security policies are:

- 0e4a0156-dd5d-11d2-8c2f-00c04fb6bcde
- 1453c42c-0fa6-11d2-a910-00c04f990f3b
- 10f24e8e-0fa6-11d2-a910-00c04f990f3b
- 1544f5e0-613c-11d1-93df-00c04fd7bd09

The corresponding TCP/UDP ports are dynamic. To permit them, you use the following statement for each number:

```
set applications application-name term term-name uuid hex-number
```

The ALG maps the program numbers into dynamically negotiated TCP/UDP ports based on these four UUIDs and permits or denies the service based on a policy you configure.

Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
15.1X49-D90	In Junos OS Release 15.1X49-D90 and earlier, on all SRX Series Firewalls, the custom application universal unique identifier (UUID) of Microsoft remote procedure call (MS-RPC) with leading zeros and the nil UUID (00000000-0000-0000-0000-000000000000) might match all TCP traffic and referenced policies allowing all TCP traffic instead of entering MS-RPC ALG check.
15.1X49-D90	In Junos OS Release 15.1X49-D90 and earlier, on all SRX Series Firewalls, the custom application universal unique identifier (UUID) of Microsoft remote procedure call (MS-RPC) with leading zeros and the nil UUID (00000000-0000-0000-0000-000000000000) might match all TCP traffic and referenced policies allowing all TCP traffic instead of entering MS-RPC ALG check.
15.1X49-D100	Starting with Junos OS Release 15.1X49-D100 and Junos OS Release 17.3R1, the custom application UUID with leading zeros does not match all TCP traffic and referenced policies, which will enter MS-RPC ALG check. This new application does not allow the nil UUID.
15.1X49-D100	Starting with Junos OS Release 15.1X49-D100 and Junos OS Release 17.3R1, the custom application UUID with leading zeros does not match all TCP traffic and referenced policies, which will enter MS-RPC ALG check. This new application does not allow the nil UUID.
15.1X49-D10	Starting in Junos OS 15.1X49-D10 and Junos OS Release 17.3R1, you can define the Sun RPC mapping entry ageout value.
15.1X49-D10	Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, you can define the MS-RPC mapping entry ageout value.

RELATED DOCUMENTATION

[RTSP ALG | 106](#)

[SQLNET ALG | 124](#)

[TALK ALG | 144](#)

RSH ALG

IN THIS SECTION

- [Understanding the RSH ALG | 87](#)
- [Example: Configuring the RSH ALG | 88](#)

The Remote Shell (RSH) provides a conduit to execute commands on a remote host. Unlike Telnet or SSH, which create a terminal shell session on the remote system, RSH passes the command and authentication data. The protocol uses the 514 TCP port to pass the authentication data and the command. The server returns the stdout of the command to the client's source port. RSH requires an ALG to pass a second client port to the server for transmission of the stderr stream.

Understanding the RSH ALG

The Remote Shell (RSH) Application Layer Gateway (ALG) processes RSH packets that initiate requests and open two gates to allow return packets from the reverse direction to the client. One gate is used for an identification (ident) session to apply authorization and the other gate is used for a standard error (stderr) session to transfer an error message.



NOTE: The RSH ALG does not work if Port Address Translation (PAT) is configured. The RSH requires the port range to be between 512 to 1024. The source NAT module cannot match this port range.

SEE ALSO

| [Understanding Data ALG Types | 13](#)

Example: Configuring the RSH ALG

IN THIS SECTION

- [Requirements | 88](#)
- [Overview | 88](#)
- [Configuration | 91](#)
- [Verification | 103](#)

This example shows how to configure the RSH ALG in route or NAT mode. The configuration allows RSH traffic to pass through a device, and it transfers remote commands and results between a client and a server located on opposite sides of a Juniper Networks device.

Requirements

This example uses the following hardware and software components:

- An SRX Series Firewall
- Two PCs (server and client)

Before you begin:

- Understand the concepts behind ALGs. See ["ALG Overview" on page 2](#)
- Understand the basics of RSH ALG. See the ["Understanding the RSH ALG" on page 87](#)

Overview

IN THIS SECTION

- [Topology | 89](#)

In this example, first you configure network interfaces on the device. Create security zones and assign interfaces to the zones, and configure a policy to allow RSH traffic to go through an SRX Series Firewall.

Then you create a static NAT rule set rs1 with a rule r1 to match with the destination address 40.0.172.10/32, and you create a static NAT prefix with address 40.0.172.45/32.

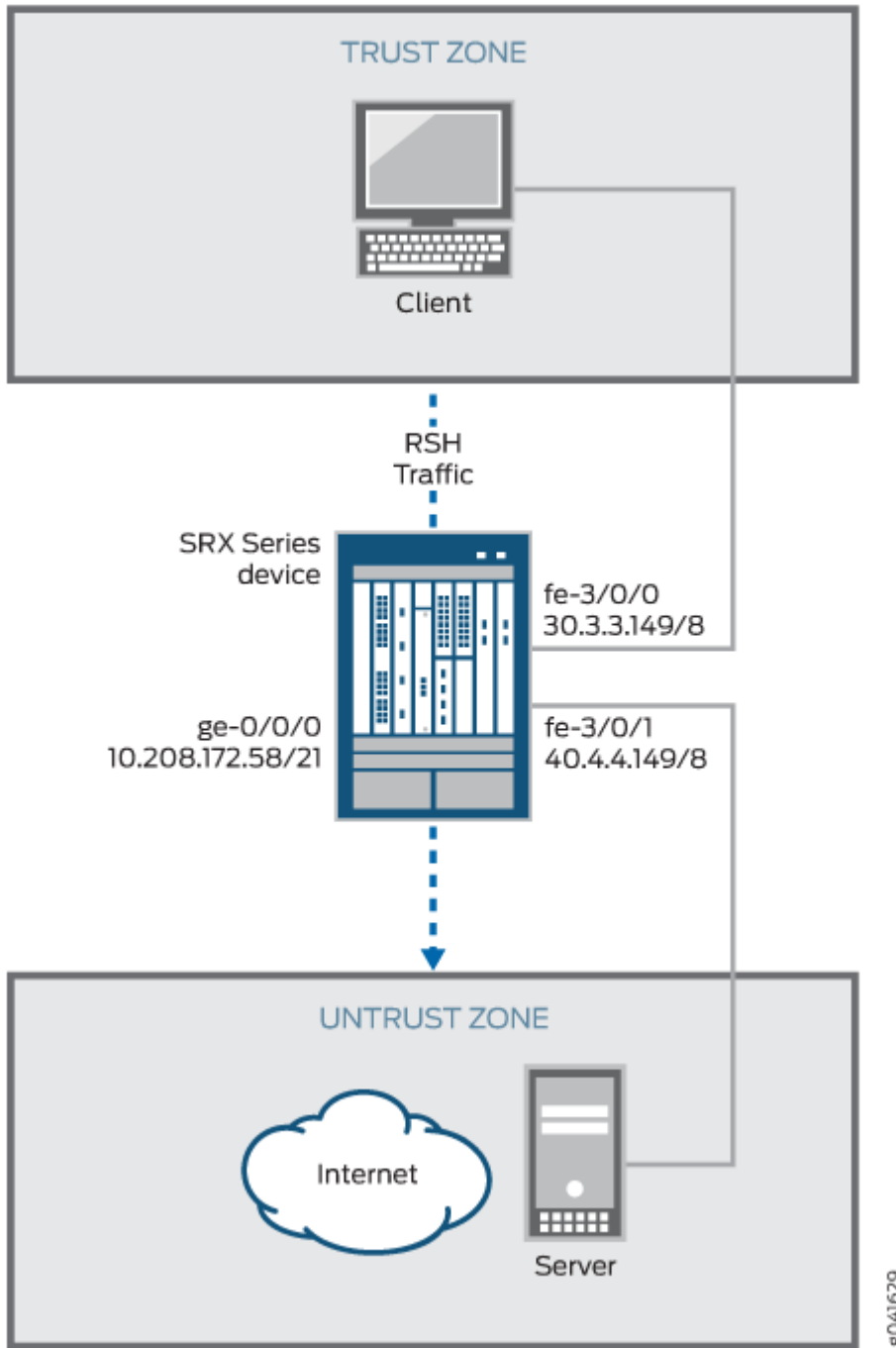
Next you create a source NAT pool src-p1 with a source rule set src-rs1 to translate packets from interface fe-3/0/0.0 to interface fe-3/0/1.0. For matching packets, the source address is translated to an IP address in the src-p1 pool.

Then you create a destination NAT pool des-p1 with a destination rule set des-rs1 to translate packets from zone trust to destination address 40.0.172.10/32. For matching packets, the destination address is translated to an IP address in the des-p1 pool. Finally, you enable RSH ALG trace options.

Topology

[Figure 5 on page 90](#) shows the RSH ALG topology.

Figure 5: RSH ALG Topology



Configuration

IN THIS SECTION

- [Configuring a Route Mode | 91](#)
- [Configuring a Static NAT Rule Set | 95](#)
- [Configuring a Source NAT Pool and Rule Set without PAT | 96](#)
- [Configuring a Destination NAT Pool and Rule Set | 99](#)
- [Enabling RSH ALG Trace Options | 101](#)

To configure the RSH ALG, perform these tasks:

Configuring a Route Mode

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.208.172.58/21
set interfaces fe-3/0/0 unit 0 family inet address 30.3.3.149/8
set interfaces fe-3/0/1 unit 0 family inet address 40.4.4.149/8
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-3/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-3/0/1.0
set security policies from-zone trust to-zone untrust policy rsh match source-address any
set security policies from-zone trust to-zone untrust policy rsh match destination-address any
set security policies from-zone trust to-zone untrust policy rsh match application junos-rsh
set security policies from-zone trust to-zone untrust policy rsh then permit
```


Step-by-Step Procedure

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

To configure route mode:

1. Configure interfaces.

```
[edit interfaces]
user@host#set ge-0/0/0 unit 0 family inet address 10.208.172.58/21
user@host#set fe-3/0/0 unit 0 family inet address 30.3.3.149/8
user@host#set fe-3/0/1 unit 0 family inet address 40.4.4.149/8
```

2. Configure zones and assign interfaces to the zones.

```
[edit security zones security-zone]
user@host#set trust host-inbound-traffic system-services all
user@host#set trust host-inbound-traffic protocols all
user@host#set trust interfaces fe-3/0/0.0
user@host#set untrust host-inbound-traffic system-services all
user@host#set untrust host-inbound-traffic protocols all
user@host#set untrust interfaces fe-3/0/0.1
```

3. Configure an RSH policy that allows RSH traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust]
user@host#set policy rsh match source-address any
user@host#set policy rsh match destination-address any
user@host#set policy rsh match application junos-rsh
user@host#set policy rsh then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example for correction.

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

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.208.172.58/21;
        }
    }
}
fe-3/0/0 {
    unit 0 {
        family inet {
            address 30.3.3.149/8;
        }
    }
}
fe-3/0/1 {
    unit 0 {
        family inet {
            address 40.4.4.149/8;
        }
    }
}
```

```
[edit]
user@host# show security zones
..
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-3/0/0.0;
```

```

    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-3/0/1.0;
    }
}
...

```

```

[edit]
user@host# show security policies
from-zone trust to-zone untrust {
    policy rsh {
        match {
            source-address any;
            destination-address any;
            application junos-rsh;
        }
        then {
            permit;
        }
    }
}
}

```

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

Configuring a Static NAT Rule Set

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security nat static rule-set rs1 from zone trust
set security nat static rule-set rs1 rule r1 match destination-address 40.0.172.10/32
set security nat static rule-set rs1 rule r1 then static-nat prefix 40.0.172.45/32
```

Step-by-Step Procedure

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

To configure a static NAT rule set:

1. Create a static NAT rule set.

```
[edit security nat static rule-set rs1]
user@host# set from zone trust
```

2. Define the rule to match with the destination address.

```
[edit security nat static rule-set rs1]
user@host# set rule r1 match destination-address 40.0.172.10/32
```

3. Define the static NAT prefix for the device.

```
[edit security nat static rule-set rs1]
user@host# set rule r1 then static-nat prefix 40.0.172.45/32
```

Results

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

```
[edit]
user@host# show security nat
static {
    rule-set rs1 {
        from zone trust;
        rule r1 {
            match {
                destination-address 40.0.172.10/32;
            }
            then {
                static-nat {
                    prefix {
                        40.0.172.45/32;
                    }
                }
            }
        }
    }
}
```

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

Configuring a Source NAT Pool and Rule Set without PAT

CLI Quick Configuration



NOTE: The RSH ALG does not support PAT configuration. The RSH ALG requires the stderr port range to be between 512 to 1024. The source NAT module cannot match this port range.

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security nat source pool src-p1 address 40.0.172.100/32 to 40.0.172.101/32
set security nat source pool src-p1 port no-translation
set security nat source rule-set src-rs1 from interface fe-3/0/0.0
set security nat source rule-set src-rs1 to interface fe-3/0/1.0
set security nat source rule-set src-rs1 rule r1 match source-address 30.0.0.0/8
set security nat source rule-set src-rs1 rule r1 match destination-address 40.0.0.0/8
set security nat source rule-set src-rs1 rule r1 then source-nat pool src-p1
```

Step-by-Step Procedure

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

To configure a source NAT pool and rule set:

1. Create a source NAT pool.

```
[edit security nat source]
user@host#set pool src-p1 address 40.0.172.100/32 to 40.0.172.101/32
```

2. Create a source NAT pool with no port translation.

```
[edit security nat source ]
set pool src-p1 port no-translation
```

3. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set src-rs1 from interface fe-3/0/0.0
user@host# set rule-set src-rs1 to interface fe-3/0/1.0
```

4. Configure a rule that matches packets and translates the source address to an address in the source pool.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule r1 match source-address 30.0.0.0/8
```

5. Configure a rule that matches packets and translates the destination address to an address in the source pool.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule r1 match destination-address 40.0.0.0/8
```

6. Configure a source NAT pool in the rule.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule r1 then source-nat pool src-p1
```

Results

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

```
[edit]
user@host# show security nat
source {
  pool src-p1 {
    address {
      40.0.172.100/32 to 40.0.172.101/32;
    }
    port no-translation;
  }
  rule-set src-rs1 {
    from interface fe-3/0/0.0;
    to interface fe-3/0/1.0;
    rule r1 {
      match {
        source-address 30.0.0.0/8;
        destination-address 40.0.0.0/8;
```


1. Create a destination NAT pool.

```
[edit security nat destination]
user@host# set pool des-p1 address 40.0.172.45/32
```

2. Create a destination NAT rule set.

```
[edit security nat destination]
user@host# set rule-set des-rs1 from zone trust
```

3. Configure a rule that matches packets and translates the source address to the address in the pool.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 match source-address 30.0.172.12/32
```

4. Configure a rule that matches packets and translates the destination address to the address in the pool.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 match destination-address 40.0.172.10/32
```

5. Configure a source NAT pool in the rule.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 then destination-nat pool des-p1
```

Results

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

```
[edit]
user@host# show security nat
destination {
  pool des-p1 {
    address {
```

```

        40.0.172.45/32;
    }
}
rule-set des-rs1 {
    from zone trust;
    rule des-r1 {
        match {
            source-address 30.0.172.12/32;
            destination-address 40.0.172.10/32;
        }
        then {
            destination-nat {
                pool {
                    des--p1;
                }
            }
        }
    }
}
}
}
}
}

```

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

Enabling RSH ALG Trace Options

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```

set security alg rsh traceoptions flag all
set security alg traceoptions file trace
set security alg traceoptions file size 1g
set security alg traceoptions level verbose

```

Step-by-Step Procedure

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

To enable RSH ALG trace options:

1. Enable RSH ALG trace options.

```
[edit security alg]
user@host#set sql traceoptions flag all
```

2. Configure a filename to receive output from the tracing operation.

```
[edit security alg]
user@host#set traceoptions file trace
```

3. Specify the maximum trace file size.

```
[edit security alg]
user@host#set traceoptions file size 1g
```

4. Specify the level of tracing output.

```
[edit security alg]
user@host#set traceoptions level verbose
```

Results

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

```
[edit]
user@host# show security alg
traceoptions {
    file trace size 1g;
    level verbose;
}
rsh traceoptions flag all;
```

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

Verification

IN THIS SECTION

- [Verifying the RSH ALG Control Session | 103](#)
- [Verifying the RSH ALG | 104](#)
- [Verifying the RSH ALG Resource Manager Group | 105](#)
- [Verifying the RSH ALG Resource Information | 105](#)

Confirm that the configuration is working properly.

Verifying the RSH ALG Control Session

Purpose

Verify that the RSH command is executed and all the RSH control and data sessions are created.

Action

From operational mode, enter the `show security flow session` command.

```
user@host>show security flow session
Session ID: 2924, Policy name: rsh/6, Timeout: 2, Valid
Resource information : RSH ALG, 2, 0
  In: 30.0.172.12/1023 --> 40.0.172.45/514;tcp, If: fe-3/0/0.0, Pkts: 7, Bytes: 320
  Out: 40.0.172.45/514 --> 30.0.172.12/1023;tcp, If: fe-3/0/1.0, Pkts: 7, Bytes: 314

Session ID: 2925, Policy name: rsh/6, Timeout: 2, Valid
Resource information : RSH ALG, 2, 24
  In: 40.0.172.45/44864 --> 30.0.172.12/113;tcp, If: fe-3/0/1.0, Pkts: 5, Bytes: 278
  Out: 30.0.172.12/113 --> 40.0.172.45/44864;tcp, If: fe-3/0/0.0, Pkts: 5, Bytes: 345

Session ID: 2926, Policy name: rsh/6, Timeout: 2, Valid
Resource information : RSH ALG, 2, 23
  In: 40.0.172.45/1023 --> 30.0.172.12/1022;tcp, If: fe-3/0/1.0, Pkts: 4, Bytes: 216
  Out: 30.0.172.12/1022 --> 40.0.172.45/1023;tcp, If: fe-3/0/0.0, Pkts: 3, Bytes: 164
Total sessions: 3
```

Meaning

- **Session ID**—Number that identifies the session. Use this ID to get more information about the session such as policy name, number of packets in and out.
- **Policy name**—Policy name that permitted the traffic.
- **In**—Incoming flow (source and destination IP addresses with their respective source and destination port numbers, session is TCP, and source interface for this session is fe-3/0/0.0).
- **Out**—Reverse flow (source and destination IP addresses with their respective source and destination port numbers, session is TCP, and destination interface for this session is fe-3/0/1.0).

Verifying the RSH ALG

Purpose

Verify that the RSH ALG is enabled.

Action

From operational mode, enter the `show security alg status` command.

```
user@host>show security alg status
ALG Status :
  PPTP      : Enabled
  RSH       : Disabled
  RTSP      : Enabled
  SCCP      : Enabled
  SIP       : Enabled
  TALK      : Enabled
  TFTP      : Enabled
  IKE-ESP   : Disabled
```



NOTE: The RSH ALG is disabled by default. To enable the RSH ALG, enter the `set security alg rsh` command in the configuration mode.

Meaning

The output shows the RSH ALG status as follows:

- Enabled—Shows the RSH ALG is enabled.
- Disabled—Shows the RSH ALG is disabled.

Verifying the RSH ALG Resource Manager Group

Purpose

Verify the total number of resource manager groups and active groups that are used by the RSH ALG.

Action

From operational mode, enter the `show security resource-manager group active` command.

```
user@host>show security resource-manager group active
Group ID 1: Application - RSH ALG
          Total groups 677, active groups 1
```

Verifying the RSH ALG Resource Information

Purpose

Verify the total number of resources and active resources that are used by the RSH ALG.

Action

From operational mode, enter the `show security resource-manager resource active` command.

```
user@host>show security resource-manager resource active
Resource ID 2: Group ID - 1, Application - RSH ALG

          Resource ID 1: Group ID - 1, Application - RSH ALG
          Total Resources 4044, active resources 2
```

SEE ALSO

[SQLNET ALG | 124](#)

[TALK ALG | 144](#)

RTSP ALG

IN THIS SECTION

- [Understanding the RTSP ALG | 106](#)
- [Understanding RTSP ALG Messages | 108](#)
- [Understanding RTSP ALG Conversation and NAT | 110](#)
- [Example: Configuring the RTSP ALG | 113](#)

The Real-Time Streaming Protocol (RTSP) controls the delivery of data with real-time properties such as audio and video. 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.

Understanding the RTSP ALG

IN THIS SECTION

- [Overview | 106](#)
- [RTSP Modes | 107](#)

Overview

RTSP (Real-Time Streaming Protocol) is an Application Layer protocol for controlling the delivery of data with real-time properties. It is similar in syntax and operation to HTTP/1.1. Unlike SIP and H.323, the purpose of RTSP is to access existing media files over the network and to control the replay of the

media. The typical communication is between a client (running RealPlayer for example) and a streaming media server. Commands include the ability to pause and play media files from the remote server.

RTSP is a control channel protocol between the media client and media server. The data channel uses a different protocol, usually Real-Time Transport Protocol (RTP) or RTP Control Protocol (RTCP).

In RTSP standard mode, the client sets up three network channels with the RTSP server when media data is delivered using RTP over UDP.

RTSP runs over TCP. RTP and RTCP run over UDP. The ports for RTP and RTCP packets are dynamically negotiated by the client and server using RTSP. Because RTP and RTCP ports are dynamic, these ports cannot be allowed by a static policy. The main purpose of introducing an RTSP ALG to a firewall is to create dynamic policy (pinhole) according to the result of client/server negotiation so that RTP and RTCP traffic can pass through.

When the client and server reside in different realms, they might not be able to determine how to route to the address of the RTP or RTCP offer given by the peer. In this case, ALG needs to be involved to do translation for the RTP or RTCP offer address and modify it in the payload.

After the connection is established, the RTSP ALG monitors the messages exchanged between the client and server, tracks the status change of the dialog, and returns all the resources it acquired to support an RTSP dialog back to the system after the dialog has completed or failed.

RTSP Modes

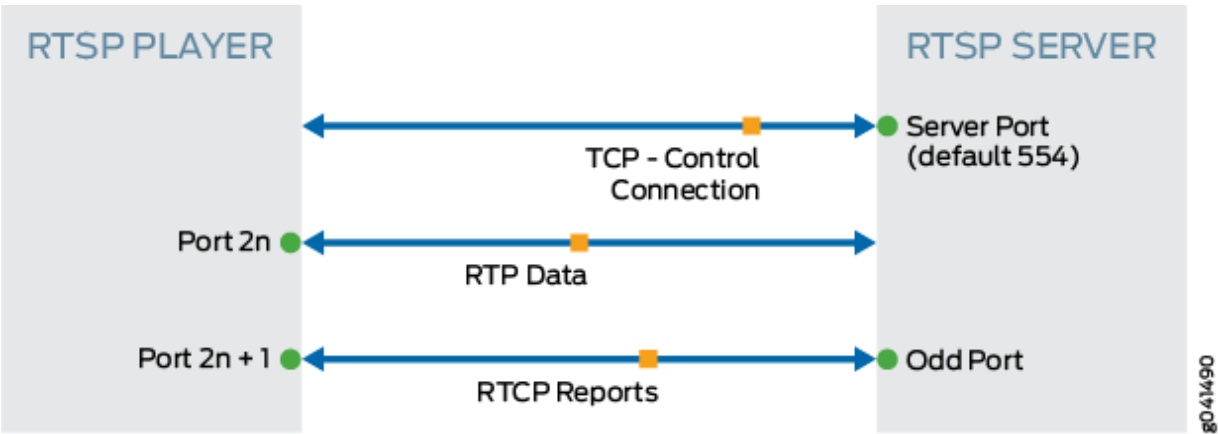
Standard Mode

In RTSP standard mode, the client sets up three network channels with the RTSP server when media data is delivered using RTP over UDP.

A full-duplex TCP connection is used for control and negotiation. A full-duplex UDP channel is used for media data delivery using the RTP packet format. In most cases, RTP is initiated from the server. A full-duplex UDP channel called RTCP is used to provide synchronization information to the client and packet loss information to the server.

[Figure 6 on page 108](#) shows the RTSP ALG standard mode.

Figure 6: RTSP ALG Standard Mode

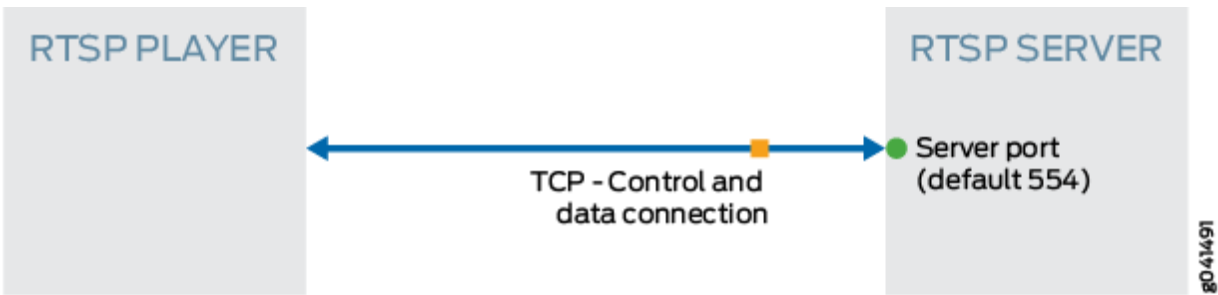


Interleave Mode

In RTSP interleave mode, media data can be made into packets using RTP or RDT over TCP. In this scenario, a single full-duplex TCP connection is used for both control and for media data delivery from the RTSP server to the client. The data stream is interleaved with the RTSP control stream.

[Figure 7 on page 108](#) shows the RTSP ALG interleave mode.

Figure 7: RTSP ALG Interleave Mode



Understanding RTSP ALG Messages

IN THIS SECTION

● RTSP Message Format | 109

- [RTSP Methods | 109](#)
- [RTSP Status Code | 110](#)
- [RTSP Header | 110](#)

RTSP Message Format

RTSP is text based and uses the ISO 10646 character set in UTF-8 encoding. Lines are terminated by CRLF, and an empty line is the separator of the message and body.

The first line is called the start-line. For request messages from client to server, the start-line represents the RTSP method. For the response message from server to client, the start-line represents the RTSP status code as the reply of method. The status code element is a 3-digit integer result code.

RTSP Methods

There are nine types of methods during one transaction.

- **OPTION**—Represents a request for information about the communication options available on the request/response chain identified by the Request-URL. This method allows the client to determine the options, requirements, or both associated with a resource, or the capabilities of a server, without implying a resource action or initiating a resource retrieval.
- **DESCRIBE**—Retrieves the description of a presentation or media object identified by the request URL from a server. This method might use the Accept header to specify the description formats that the client interprets.
- **ANNOUNCE**—Request sent from client to server, this method posts the description of a presentation or media object identified by the request URL to a server. When request sent from server to client, this method updates the session description in real-time.
- **SETUP**—Requests a URI and specifies the transport mechanism to be used for the streamed media.
- **PLAY**—Informs the server to start sending data using the mechanism specified in SETUP.
- **PAUSE**—Requests the stream delivery to be interrupted temporarily.
- **TEARDOWN**—Stops the stream delivery for the given URI, freeing the resource associated with it.
- **GET_PARAMETER**—Retrieves the value of a parameter of a presentation or stream specified in the URI.
- **SET_PARAMETER**—Sets the value of a parameter for a presentation or stream specified by the URI.

RTSP Status Code

The first digit of the status code defines the class of response.

- **1****: Informational—Request received, continuing process.
- **2****: Success
- **3****: Redirection—Further action must be taken in order to complete the request.
- **4****: Client Error—The request contains bad syntax or cannot be fulfilled.
- **5****: Server Error—The server failed to fulfill an apparently valid request.

RTSP Header

The RTSP header consists of the following fields:

- **CSeq**—Specifies the sequence number for an RTSP request-response pair. For every RTSP request containing the given sequence number, there will be a corresponding response having the same number.
- **Content-Length**—Contains the length of the content of the method, that is, after the double CRLF following the last header.
- **TRANSPORT**—Indicates which transport protocol is to be used and configures its parameters.
- **SESSION**—Identifies an RTSP session started by the media server in a SETUP response and concluded by TEARDOWN on the presentation URL.

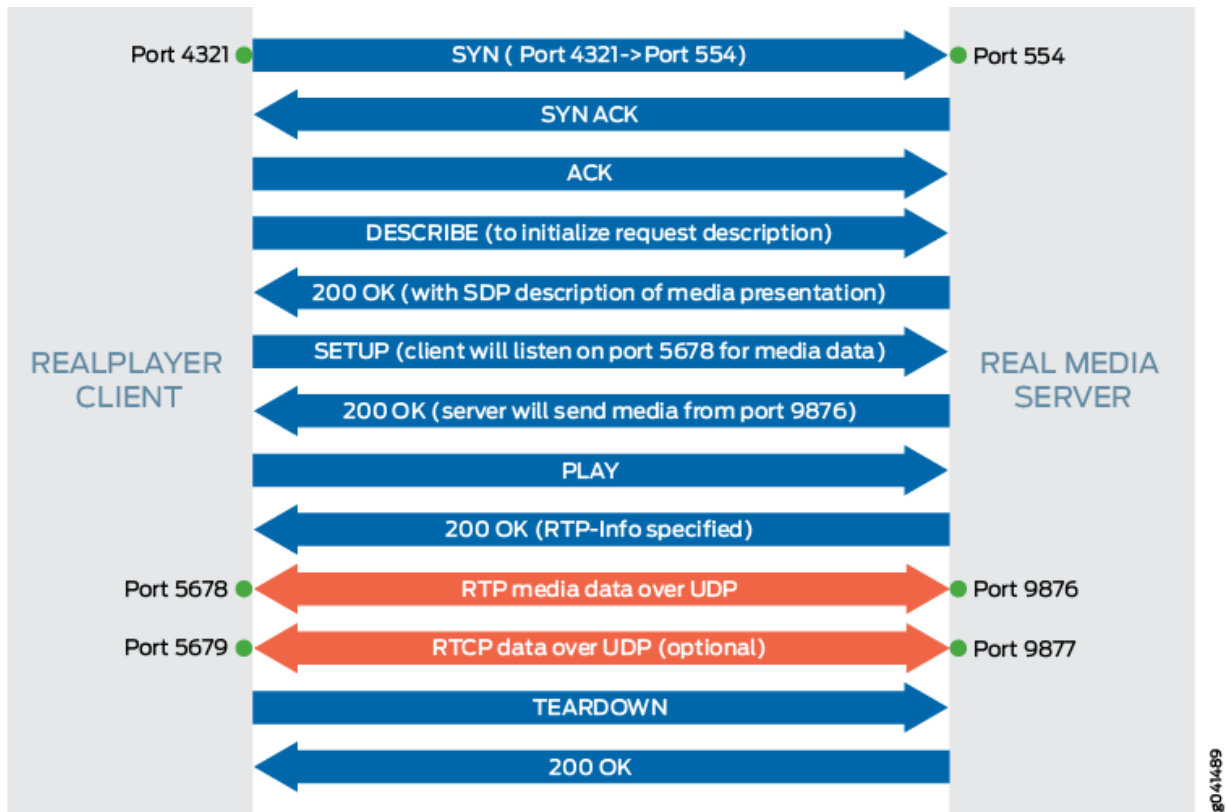
Understanding RTSP ALG Conversation and NAT

This topic provides details on typical RTSP ALG conversation.

In general, RTP and RTCP packets are bidirectional, which means that either the client or server could initiate an RTP or an RTCP session.

[Figure 8 on page 111](#) describes an example of a sample packet capture in a standard RTSP conversation.

Figure 8: RTSP ALG Conversation



The RTSP ALG performs the following actions for a RTSP sample packet capture in a standard RTSP conversation:

1. Monitors SETUP and 200 OK messages.
2. Receives negotiated ports (6543 and 8765 in this example)
3. Opens a pinhole for UDP media data from server to client.
4. Receives the IP address in payload and translates the address if NAT is required.

[Table 4 on page 112](#) describes the RTSP payload IP NAT.

Table 4: RTSP Payload IP NAT

	Forward(C->S)	Reverse(S->C)	Pinhole	Payload IP Translate	Payload Port Translate
No NAT	A/4321->B/554	A/4321<-B/554	B/9876->A/5678 A/5678->B/9876	N/A	N/A
Source NAT (IPv _x)	A/4321->B/554	A'/P'<-B/554	B/9876->A'/P'' A/5678->B/9876	N/A (*)	5678<->P''
Destination NAT (IPv _x)	A/4321->B'/554	A/4321<-B/554	B/9876->A/5678 A/5678->B'/9876	B' -> B (**)	N/A
NAT64	A/4321->B''/554	A''/Q'<-B/554	B/9876->A''/Q'' A/5678->B''/9876	B''(IPv6)->B(IPv4)	5678<->Q''
NAT46	A/4321->B'''/554	A'''/R'<-B/554	B/9876->A'''/R'' A/5678->B'''/9876	B'''(IPv4)->B(IPv6)	5678<->R''

In [Table 4 on page 112](#), the following letters and symbols are used:

- A—RTSP client IP address
- A'—Translated IPv4 or IPv6 address of RTSP client
- A''—Translated IPv4 address
- A'''—Translated IPv6 address
- B—RTSP server IP address
- B'—RTSP server IP address before destination NAT
- B''—RTSP server IP address at IPv6 realm
- B'''—RTSP server IP address at IPv4 realm
- P'—Translated Port(translates from 4321) of RTSP client

- P'—Translated Port(translates from 5678 in message payload) of RTSP client
- Q'—Translated (IPv6 to IPv4) Port(translates from 4321) of RTSP client
- Q''—Translated (IPv6 to IPv4) Port (translates from 5678 in message payload) of RTSP client
- R'—Translated (IPv4 to IPv6) Port (translates from 4321) of RTSP client
- R''—Translated (IPv4 to IPv6) Port (translates from 5678 in message payload) of RTSP client
- (*)—RTSP server IP address B appears in payload message; it does not need to translate
- (**)—IP address B' appears in payload message from client to server; it needs to translate to B

Example: Configuring the RTSP ALG

IN THIS SECTION

- [Requirements | 113](#)
- [Overview | 113](#)
- [Configuration | 114](#)
- [Verification | 119](#)

This example shows how to configure the RTSP ALG to pass through RTSP traffic with a source NAT pool on Juniper Networks devices.

Requirements

- Configure proxy ARP for all IP addresses in the source NAT pool.
- Enable the RTSP ALG.
- Understand the basics concepts of the RTSP ALG. See "[Understanding the RTSP ALG](#)" on page 106.

Overview

In this example, the RTSP ALG is configured to monitor and allow RTSP traffic transferring media between client and server located on opposite sides of a Juniper Networks device.

Configuration

IN THIS SECTION

- [Enabling RTSP ALG | 114](#)
- [Configuring a NAT Source Pool and Rule Set and a Policy | 114](#)
- [Configuring RTSP ALG trace options | 117](#)

Enabling RTSP ALG

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

To configure proxy ARP for all IP addresses in the source NAT pool and to enable RTSP ALG:

```
set security nat proxy-arp interface <interface-name> address 10.10.10.1/32 to 10.10.10.10/32
set security alg rtsp
```

Enter `commit` from configuration mode.

Configuring a NAT Source Pool and Rule Set and a Policy

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security nat source pool pool1 address 10.10.10.1/32 to 10.10.10.10/32
set security zones security-zone green address-book address sa1 1.1.1.0/24
set security zones security-zone red address-book address da1 2.2.2.0/24
```

```
set security nat source rule-set rs1 from zone green
set security nat source rule-set rs1 to zone red
set security nat source rule-set rs1 rule r1 match source-address 1.1.1.0/24
set security nat source rule-set rs1 rule r1 match destination-address 2.2.2.0/24
set security nat source rule-set rs1 rule r1 then source-nat pool pool1
```

```
set security policy from-zone green to-zone red policy pol1 match destination-address da1
set security policy from-zone green to-zone red policy pol1 match source-address sa1
set security policy from-zone green to-zone red policy pol1 match application junos-rtsp
set security policy from-zone green to-zone red policy pol1 then permit
```

Enter `commit` from configuration mode.



NOTE: If you are not sure of the RTSP client and server IP address, you can replace “da1” and “sa1” with “any”.

Step-by-Step Procedure

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

To configure a source NAT pool:

1. Create a NAT source pool.

```
[edit security]
user@host# set nat source pool pool1 address 10.10.10.1/32 to 10.10.10.10/32
```

2. Configure security zone address book entries.

```
[edit security zones security-zone]
user@host# set green address-book address sa1 1.1.1.0/24
user@host# set red address-book address da1 2.2.2.0/24
```

3. Create a NAT source rule set.

```
[edit security nat source rule-set rs1]
user@host# set from zone green
```



```

user@host# set to zone red
user@host# set rule r1 match source-address 1.1.1.0/24
user@host# set rule r1 match destination-address 2.2.2.0/24
user@host# set rule r1 then source-nat pool pool1

```

4. Configure a policy.

```

[edit security policies from-zone green to-zone red policy pol1]
user@host# set match source-address sa1
user@host# set match destination-address da1
user@host# set match application junos-rtsp
user@host# set then permit

```

Results

From configuration mode, confirm your configuration by entering the `show security nat` and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit ]
user@host# show security nat
source {
    pool pool1 {
        address {
            10.10.10.1/32 to 10.10.10.10/32;
        }
    }
}
rule-set rs1 {
    from zone green;
    to zone red;
    rule r1 {
        match {
            source-address 1.1.1.0/24;
            destination-address 2.2.2.0/24;
        }
        then {
            source-nat {
                pool {
                    pool1;
                }
            }
        }
    }
}

```

```

    }
  }
}

```

```

[edit]
user@host# show security policies
from-zone green to-zone red {policy pol1 {
  policy pol1 {
    match {
      source-address sa1;
      destination-address da1;
      application [junos-rtsp];
    }
    then {
      permit;
    }
  }
}
default-policy {
  permit-all;
}

```

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

Configuring RTSP ALG trace options

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```

set security alg rtsp traceoptions flag all
set security alg traceoptions file trace
set security alg traceoptions file size 1g
set security alg traceoptions level verbose

```

Step-by-Step Procedure

To configure RTSP ALG trace options:

1. Enable RTSP ALG trace options.

```
[edit security alg]
user@host# set rtsp traceoptions flag all
```

2. Configure a filename to receive output from the tracing operation.

```
[edit security alg]
user@host# set traceoptions file trace
```

3. Specify the maximum trace file size.

```
[edit security alg]
user@host# set traceoptions file size 1g
```

4. Specify the level of tracing output.

```
[edit security alg]
user@host# set traceoptions level verbose
```

Results

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

```
[edit]
user@host# show security alg
traceoptions {
    file trace size 1g;
    level verbose;
}
rtsp traceoptions flag all;
```

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

Verification

IN THIS SECTION

- [Verifying RTSP ALG | 119](#)
- [Verifying the RTSP ALG Control Session | 120](#)
- [Verifying the RTSP ALG Flow Gate Information | 121](#)
- [Verifying the RTSP Resource Manager Group | 122](#)
- [Verifying the RTSP Resource Information | 123](#)

Confirm that the configuration is working properly.

Verifying RTSP ALG

Purpose

Verify that the RTSP ALG is enabled.

Action

From operational mode, enter the `show security alg status` command.

```
user@host> show security alg status
```

```
DNS      : Enabled
FTP       : Enabled
H323     : Enabled
RTSP     : Enabled
```

Meaning

The output shows the RTSP ALG status as follows:

- Enabled—Shows the RTSP ALG is enabled.

- Disabled—Shows the RTSP ALG is disabled.

Verifying the RTSP ALG Control Session

Purpose

Verify that the control session is created and all the RTSP control and data sessions are created.

Action

From operational mode, enter the `show security flow session` command.

```
user@host>show security flow session
Flow Sessions on FPC5 PIC0:

Session ID: 100004087, Policy name: dns-alg/4, Timeout: 1798, Valid
Resource information : RTSP ALG, 1, 0
  In: 1.1.0.100/59889 --> 1.1.0.202/554;tcp, If: ge-0/0/1.0, Pkts: 28, Bytes: 7618
  Out: 1.1.0.202/554 --> 1.1.0.100/59889;tcp, If: ge-0/0/2.0, Pkts: 27, Bytes: 24304

Session ID: 100004088, Policy name: dns-alg/4, Timeout: 120, Valid
Resource information : RTSP ALG, 1, 1
  In: 1.1.0.202/5004 --> 1.1.0.100/62092;udp, If: ge-0/0/2.0, Pkts: 19, Bytes: 17013
  Out: 1.1.0.100/62092 --> 1.1.0.202/5004;udp, If: ge-0/0/1.0, Pkts: 0, Bytes: 0

Session ID: 100004089, Policy name: dns-alg/4, Timeout: 120, Valid
Resource information : RTSP ALG, 1, 4
  In: 1.1.0.202/5004 --> 1.1.0.100/62094;udp, If: ge-0/0/2.0, Pkts: 433, Bytes: 346183
  Out: 1.1.0.100/62094 --> 1.1.0.202/5004;udp, If: ge-0/0/1.0, Pkts: 0, Bytes: 0

Session ID: 100004090, Policy name: dns-alg/4, Timeout: 120, Valid
Resource information : RTSP ALG, 1, 3
  In: 1.1.0.100/62093 --> 1.1.0.202/5005;udp, If: ge-0/0/1.0, Pkts: 2, Bytes: 260
  Out: 1.1.0.202/5005 --> 1.1.0.100/62093;udp, If: ge-0/0/2.0, Pkts: 0, Bytes: 0
Total sessions: 4
```

Meaning

- **Session ID**—Number that identifies the session. Use this ID to get more information about the session such as policy name or number of packets in and out.

- **Policy name**—Policy name that permitted the traffic.
- **In**—Incoming flow (source and destination IP addresses with their respective source and destination port numbers, session is TCP, and the source interface for this session is ge-0/0/1.0).
- **Out**—Reverse flow (source and destination IP addresses with their respective source and destination port numbers, session is TCP, and destination interface for this session is fe-0/0/2.0).

Verifying the RTSP ALG Flow Gate Information

Purpose

Verify that the flow gate is opened for TCP data channel connection.

Action

From operational mode, enter the `show security flow gate` command.

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

```
Flow Gates on FPC5 PIC0:
```

```
Hole: 1.1.0.202-1.1.0.202/5005-5005->1.1.0.100-1.1.0.100/62093-62093
```

```
Translated: 0.0.0.0/0->0.0.0.0/0
```

```
Protocol: udp
```

```
Application: RTSP ALG/11
```

```
Age: 32 seconds
```

```
Flags: 0x0080
```

```
Zone: untrust
```

```
Reference count: 1
```

```
Resource: 4-1-2
```

```
Hole: 1.1.0.100-1.1.0.100/62093-62093->1.1.0.202-1.1.0.202/5005-5005
```

```
Translated: 0.0.0.0/0->0.0.0.0/0
```

```
Protocol: udp
```

```
Application: RTSP ALG/11
```

```
Age: 32 seconds
```

```
Flags: 0x0080
```

```
Zone: trust
```

```
Reference count: 1
```

```
Resource: 4-1-3
```

```
Hole: 1.1.0.202-1.1.0.202/5004-5004->1.1.0.100-1.1.0.100/62094-62094
Translated: 0.0.0.0/0->0.0.0.0/0
Protocol: udp
Application: RTSP ALG/11
Age: 32 seconds
Flags: 0x0080
Zone: untrust
Reference count: 1
Resource: 4-1-4
```

```
Hole: 1.1.0.100-1.1.0.100/62094-62094->1.1.0.202-1.1.0.202/5004-5004
Translated: 0.0.0.0/0->0.0.0.0/0
Protocol: udp
Application: RTSP ALG/11
Age: 32 seconds
Flags: 0x0080
Zone: trust
Reference count: 1
Resource: 4-1-5
```

```
Valid gates: 4
Pending gates: 0
Invalidated gates: 0
Gates in other states: 0
Total gates: 4
```

Meaning

The sample output shows that the flow gate is opened for TCP data channel connection.

Verifying the RTSP Resource Manager Group

Purpose

Verify the total number of resource manager groups and active groups that are used by the RTSP ALG.

Action

From operational mode, enter the `show security resource-manager group active` command.

```
user@host>show security resource-manager group active
Group ID 1: Application - RTSP ALG
Total groups 19763, active groups 1
```

Meaning

The sample output shows the total number of resource manager groups and active groups that are used by the RTSP ALG.

Verifying the RTSP Resource Information

Purpose

Verify the total number of resources and active resources that are used by the RTSP ALG.

Action

From operational mode, enter the `show security resource-manager resource active` command.

```
user@host>show security resource-manager resource active
Resource ID 2: Group ID - 1, Application - RTSP ALG

Resource ID 1: Group ID - 1, Application - RTSP ALG
Total Resources 93286, active resources 2
```

Meaning

The sample output shows the total number of resources and active resources that are used by the RTSP ALG.

RELATED DOCUMENTATION

[TALK ALG | 144](#)

[TFTP ALG | 163](#)

SQLNET ALG

IN THIS SECTION

- [Understanding the SQLNET ALG | 124](#)
- [Example: Configuring the SQLNET ALG | 125](#)

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.

Understanding the SQLNET ALG

The SQLNET Application Layer Gateway (ALG) processes Transparent Network Substrate (TNS) REDIRECT packets for IP addresses and port information. The SQLNET ALG performs Network Address Translation (NAT) on the payload of the TNS REDIRECT packet, opens a pinhole for a new connection from a client to a server, and transfers data between a client and a server located on opposite sides of a Juniper Networks device.

SQLNET ALG supports the following types of data transfer modes:

- Redirect mode — connect-redirect type
- Interleave mode — connect-accept type
- Load balance — connect-redirect-connect-redirect type

SQLNET allows remote data access between applications and the Oracle database, or among multiple Oracle databases. SQLNET primarily establishes and maintains connection between a client application and an Oracle database server. SQLNET has several communication layers that enable clients and database servers to share, modify, and manipulate data.

Oracle SQL servers use the SQLNET protocol to execute SQL commands from clients, including load balancing and application-specific services. The SQLNET protocol uses TNS as its networking architecture, and all SQLNET traffic is encapsulated into TNS packet format.

The SQLNET ALG monitors control packets, opens pinhole for data traffic, and performs NAT and port rewrites. Support of stateful firewall and NAT services are required to configure the SQLNET ALG for TCP port 1521.

Example: Configuring the SQLNET ALG

IN THIS SECTION

- [Requirements | 125](#)
- [Overview | 126](#)
- [Configuration | 128](#)
- [Verification | 140](#)

The SQLNET ALG processes TNS REDIRECT packets, performs NAT, and opens a pinhole for a new connection from a client to a server.

This example shows how to configure the SQLNET ALG in route or NAT mode, allow SQLNET traffic to pass through a device, and transfer data between a client and a server located on opposite sides of a Juniper Networks device.

Requirements

This example uses the following hardware and software components:

- An SRX Series Firewall
- Two PCs (client and server)

Before you begin:

- Understand the concepts behind ALGs. See ["ALG Overview" on page 2](#).
- Understand the basics of SQLNET ALG . See ["Understanding the SQLNET ALG" on page 124](#).

Overview

IN THIS SECTION

- [Topology](#) | 126

In this example, first you configure network interfaces on the device. Create security zones and assign interfaces to the zones, and configure a policy to allow SQLNET traffic to go through an SRX Series Firewall.

Then you create a static NAT rule set rs1 with a rule r1 to match with the destination address 40.0.172.10/32, and you create a static NAT prefix with address 40.0.172.45/32.

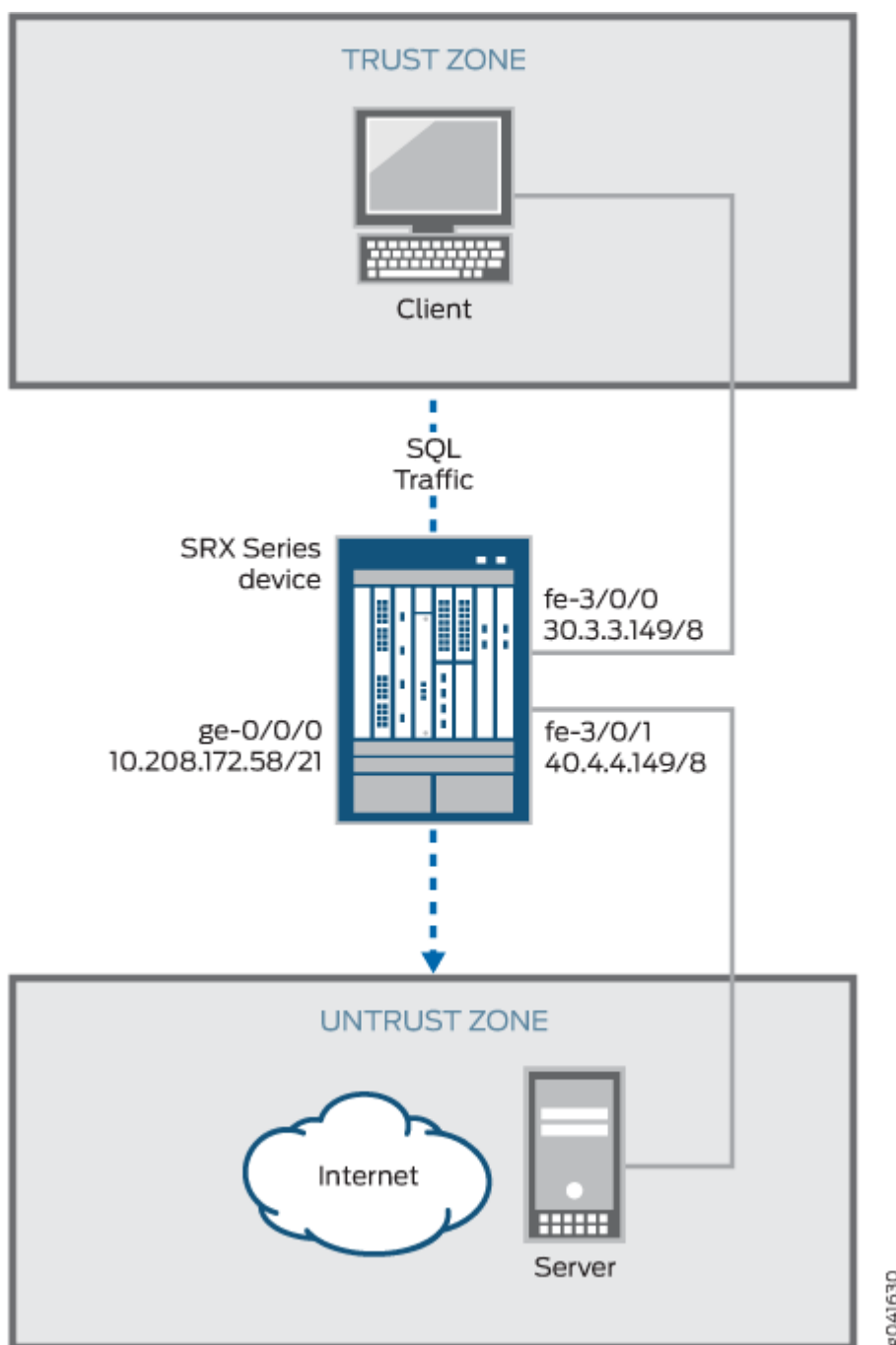
Next you create a source NAT pool src-p1 with a source rule set src-rs1 to translate packets from interface fe-3/0/0.0 to interface fe-3/0/1.0. For matching packets, the source address is translated to an IP address in the src-p1 pool.

Then you create a destination NAT pool des-p1 with a destination rule set des-rs1 to translate packets from zone trust to destination address 40.0.172.10/32. For matching packets, the destination address is translated to an IP address in the des-p1 pool. Finally, you enable SQLNET ALG trace options.

Topology

[Figure 9 on page 127](#) shows the SQLNET ALG topology.

Figure 9: SQLNET ALG Topology



Configuration

IN THIS SECTION

- [Configuring a Route Mode | 128](#)
- [Configuring a Static NAT Rule Set | 132](#)
- [Configuring a Source NAT Pool and Rule Set | 133](#)
- [Configuring a Destination NAT Pool and Rule Set | 136](#)
- [Enabling SQLNET ALG | 138](#)
- [Enabling SQLNET ALG Trace Options | 139](#)

To configure the SQLNET ALG, perform these tasks:

Configuring a Route Mode

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.208.172.58/21
set interfaces fe-3/0/0 unit 0 family inet address 30.3.3.149/8
set interfaces fe-3/0/1 unit 0 family inet address 40.4.4.149/8
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces fe-3/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-3/0/1.0
set security policies from-zone trust to-zone untrust policy sql match source-address any
set security policies from-zone trust to-zone untrust policy sql match destination-address any
set security policies from-zone trust to-zone untrust policy sql match application junos-sqlnet-
v2
set security policies from-zone trust to-zone untrust policy sql then permit
```

Step-by-Step Procedure

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

To configure route mode:

1. Configure interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet address 10.208.172.58/21
user@host# set fe-3/0/0 unit 0 family inet address 30.3.3.149/8
user@host# set fe-3/0/1 unit 0 family inet address 40.4.4.149/8
```

2. Configure zones and assign interfaces to the zones.

```
[edit security zones security-zone]
user@host# set trust host-inbound-traffic system-services all
user@host# set trust host-inbound-traffic protocols all
user@host# set trust interfaces fe-3/0/0.0
user@host# set untrust host-inbound-traffic system-services all
user@host# set untrust host-inbound-traffic protocols all
user@host# set untrust interfaces fe-3/0/1.0
```

3. Configure a SQL policy that allows SQL traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone trust to-zone untrust ]
user@host# set policy sql match source-address any
user@host# set policy sql match destination-address any
user@host# set policy sql match application junos-sqlnet-v2
user@host# set policy sql then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

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

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.208.172.58/21;
        }
    }
}
fe-3/0/0 {
    unit 0 {
        family inet {
            address 30.3.3.149/8;
        }
    }
}
fe-3/0/1 {
    unit 0 {
        family inet {
            address 40.4.4.149/8;
        }
    }
}
```

```
[edit]
user@host# show security zones
...
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-3/0/0.0;
```

```

    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        fe-3/0/1.0;
    }
}
...

```

```

[edit]
user@host# show security policies
from-zone trust to-zone untrust {
    policy sql {
        match {
            source-address any;
            destination-address any;
            application junos-sqlnet-v2;
        }
        then {
            permit;
        }
    }
}
}

```

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

Configuring a Static NAT Rule Set

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security nat static rule-set rs1 from zone trust
set security nat static rule-set rs1 rule r1 match destination-address 40.0.172.10/32
set security nat static rule-set rs1 rule r1 then static-nat prefix 40.0.172.45/32
```

Step-by-Step Procedure

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

To configure a static NAT rule set:

1. Create a static NAT rule set.

```
[edit security nat static rule-set rs1]
user@host# set from zone trust
```

2. Define a rule to match with the destination address.

```
[edit security nat static rule-set rs1]
user@host# set rule r1 match destination-address 40.0.172.10/32
```

3. Define a static NAT prefix for the device.

```
[edit security nat static rule-set rs1]
user@host# set rule r1 then static-nat prefix 40.0.172.45/32
```

Results

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

```
[edit]
user@host# show security nat
static {
    rule-set rs1 {
        from zone trust;
        rule r1 {
            match {
                destination-address 40.0.172.10/32;
            }
            then {
                static-nat {
                    prefix {
                        40.0.172.45/32;
                    }
                }
            }
        }
    }
}
```

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

Configuring a Source NAT Pool and Rule Set

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security nat source pool src-p1 address 40.0.172.100/32 to 40.0.172.101/32
set security nat source rule-set src-rs1 from interface fe-3/0/0.0
set security nat source rule-set src-rs1 to interface fe-3/0/1.0
set security nat source rule-set src-rs1 rule r1 match source-address 30.0.0.0/8
```

```
set security nat source rule-set src-rs1 rule r1 match destination-address 40.0.0.0/8
set security nat source rule-set src-rs1 rule r1 then source-nat pool src-p1
```

Step-by-Step Procedure

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

To configure a source NAT pool and rule set:

1. Create a source NAT pool.

```
[edit security nat source]
user@host# set pool src-p1 address 40.0.172.100/32 to 40.0.172.101/32
```

2. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set src-rs1 from interface fe-3/0/0.0
user@host# set rule-set src-rs1 to interface fe-3/0/1.0
```

3. Configure a rule that matches packets and translates the source address to an address in the source pool.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule r1 match source-address 30.0.0.0/8
```

4. Configure a rule that matches packets and translates the destination address to an address in the source pool.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule r1 match destination-address 40.0.0.0/8
```

5. Configure a source NAT pool in the rule.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule r1 then source-nat pool src-p1
```

Results

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

```
[edit]
user@host# show security nat
source {
  pool src-p1 {
    address {
      40.0.172.100/32 to 40.0.172.101/32;
    }
  }
  rule-set src-rs1 {
    from interface fe-3/0/0.0;
    to interface fe-3/0/1.0;
    rule r1 {
      match {
        source-address 30.0.0.0/8;
        destination-address 40.0.0.0/8;
      }
      then {
        source-nat {
          pool {
            src-p1;
          }
        }
      }
    }
  }
}
```

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

Configuring a Destination NAT Pool and Rule Set

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security nat destination pool des-p1 address 40.0.172.45/32
set security nat destination rule-set des-rs1 from zone trust
set security nat destination rule-set des-rs1 rule des-r1 match source-address 30.0.172.12/32
set security nat destination rule-set des-rs1 rule des-r1 match destination-address
40.0.172.10/32
set security nat destination rule-set des-rs1 rule des-r1 then destination-nat pool des-p1
```

Step-by-Step Procedure

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

To configure a destination NAT pool and rule set:

1. Create a destination NAT pool.

```
[edit security nat destination]
user@host# set pool des-p1 address 40.0.172.45/32
```

2. Create a destination NAT rule set.

```
[edit security nat destination]
user@host# set rule-set des-rs1 from zone trust
```

3. Configure a rule that matches packets and translates the source address to the address in the pool.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 match source-address 30.0.172.12/32
```

4. Configure a rule that matches packets and translates the destination address to the address in the pool.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 match destination-address 40.0.172.10/32
```

5. Configure a source NAT pool in the rule.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 then destination-nat pool des-p1
```

Results

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

```
[edit]
user@host# show security nat
destination {
  pool des-p1 {
    address {
      40.0.172.45/32;
    }
  }
  rule-set des-rs1 {
    from zone trust;
    rule des-r1 {
      match {
        source-address 30.0.172.12/32;
        destination-address 40.0.172.10/32;
      }
      then {
        destination-nat {
          pool {
            des-p1;
          }
        }
      }
    }
  }
}
```

```
}
}
```

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

Enabling SQLNET ALG

CLI Quick Configuration



NOTE: Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, the SQLNET application layer gateway is enabled by default.

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set security alg sql
```

Step-by-Step Procedure

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

To enable SQLNET ALG:

1. Enable SQLNET ALG.

```
[edit ]
user@host#set security alg sql
```

Enabling SQLNET ALG Trace Options

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security alg sql traceoptions flag all
set security alg traceoptions file trace
set security alg traceoptions file size 1g
set security alg traceoptions level verbose
```

Step-by-Step Procedure

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

To enable SQLNET ALG trace options:

1. Enable SQLNET ALG trace options.

```
[edit security alg]
user@host#set sql traceoptions flag all
```

2. Configure a filename to receive output from the tracing operation.

```
[edit security alg]
user@host#set traceoptions file trace
```

3. Specify the maximum trace file size.

```
[edit security alg]
user@host#set traceoptions file size 1g
```


4. Specify the level of tracing output.

```
[edit security alg]
user@host#set traceoptions level verbose
```

Results

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

```
[edit]
user@host# show security alg
  traceoptions {
    file trace size 1g;
    level verbose;
  }
sql traceoptions flag all;
```

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

Verification

IN THIS SECTION

- [Verifying the SQLNET ALG Control Session | 141](#)
- [Verifying the SQLNET ALG | 141](#)
- [Verifying the SQLNET ALG Resource Manager Group | 142](#)
- [Verifying the SQLNET ALG Resource Information | 143](#)

Confirm that the configuration is working properly.

Verifying the SQLNET ALG Control Session

Purpose

Verify that the SQL command is executed and all the SQL control and data sessions are created.

Action

From operational mode, enter the show security flow session command.

```
user@host>show security flow session
Session ID: 10880, Policy name: sql, Timeout: 2, Valid
  In: 30.0.172.12/52315 --> 40.0.172.35/1521;tcp, If: fe-3/0/0.0, Pkts: 6, Bytes: 492
  Out: 40.0.172.35/1521 --> 30.0.172.12/52315;tcp, If: fe-3/0/1.0, Pkts: 4, Bytes: 227

Session ID: 10881, Policy name: sql, Timeout: 1800, Valid
Resource information : SQLV2 ALG, 5, 18
  In: 30.0.172.12/45944 --> 40.0.172.35/1114;tcp, If: fe-3/0/0.0, Pkts: 18, Bytes: 4240
  Out: 40.0.172.35/1114 --> 30.0.172.12/45944;tcp, If: fe-3/0/1.0, Pkts: 15, Bytes: 3989
Total sessions: 2
```

Meaning

- **Session ID**—Number that identifies the session. Use this ID to get more information about the session such as policy name, number of packets in and out.
- **Policy name**—Policy name that permitted the traffic.
- **In**—Incoming flow (source and destination IP addresses with their respective source and destination port numbers, session is TCP, and source interface for this session is fe-3/0/0.0).
- **Out**—Reverse flow (source and destination IP addresses with their respective source and destination port numbers, session is TCP, and destination interface for this session is fe-3/0/1.0).

Verifying the SQLNET ALG

Purpose

Verify that the SQLNET ALG is enabled.

Action

From operational mode, enter the `show security alg status` command.

```
user@host>show security alg status
```

ALG Status :

DNS	: Enabled
FTP	: Enabled
H323	: Disabled
MGCP	: Disabled
MSRPC	: Enabled
PPTP	: Enabled
RSH	: Disabled
RTSP	: Disabled
SCCP	: Disabled
SIP	: Disabled
SQL	: Enabled
SUNRPC	: Enabled
TALK	: Enabled
TFTP	: Enabled
IKE-ESP	: Disabled

Meaning

The output shows the SQLNET ALG status as follows:

- Enabled—Shows the SQLNET ALG is enabled
- Disabled—Shows the SQLNET ALG is disabled.

Verifying the SQLNET ALG Resource Manager Group

Purpose

Verify the total number of resource manager groups and active groups that are used by the SQLNET ALG.

Action

From operational mode, enter the `show security resource-manager group active` command.

```
user@host>show security resource-manager group active
Group ID 1: Application - SQL ALG
          Total groups 677, active groups 1
```

Verifying the SQLNET ALG Resource Information

Purpose

Verify the total number of resources and active resources that are used by the SQLNET ALG.

Action

From operational mode, enter the `show security resource-manager resource active` command.

```
user@host>show security resource-manager resource active
Resource ID 2: Group ID - 1, Application - SQL ALG

          Resource ID 1: Group ID - 1, Application - SQL ALG
          Total Resources 4044, active resources 2
```

Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
15.1X49-D10	Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, the SQLNET application layer gateway is enabled by default.

RELATED DOCUMENTATION

TFTP ALG 163
DNS ALG 15
FTP ALG 28

TALK ALG

IN THIS SECTION

- [Understanding the TALK ALG | 144](#)
- [Example: Configuring the TALK ALG | 144](#)

The TALK ALG is a visual communication program used for interactive communication between two users. The TALK protocol uses UDP port 517 and port 518 for control channel connections. The talk program consists of a server and a client. The server handles client notifications and helps to establish talk sessions. There are two types of talk servers: `ntalk` and `talkd`. The TALK ALG processes packets of both `ntalk` and `talkd` formats. It also performs NAT and gate opening as necessary.

Understanding the TALK ALG

The TALK ALG is a visual communication program used for interactive communication between two users. The TALK ALG processes TALK packets, performs Network Address Translation (NAT), and opens two gates (TCP and UDP) on the receiver side. One gate is used for the next LOOKUP packet. The other gate is used for make a connection from a client to a server and to initiate communication between a client and a server located on opposite sides of a Juniper Networks device.

There are two types of TALK servers: `ntalkd` and `talkd`.

The TALK ALG processes both `ntalk` and `talkd` packets. The TALK ALG uses port UDP517 and port UDP518 to establish a connection between a client and a server.

Example: Configuring the TALK ALG

IN THIS SECTION

- [Requirements | 145](#)

- [Overview | 145](#)
- [Configuration | 148](#)
- [Verification | 159](#)

This example shows how to configure the TALK ALG in route or NAT mode, allow the TALK traffic to pass through a device, and initiate communication between a client and a server located on opposite sides of a Juniper Networks device.

Requirements

This example uses the following hardware and software components:

- An SRX Series Firewall
- Two PCs (client and server)

Before you begin:

- Understand the concepts behind ALGs. See ["ALG Overview" on page 2](#).
- Understand the basics of TALK ALG. See ["Understanding the TALK ALG" on page 144](#).

Overview

IN THIS SECTION

- [Topology | 146](#)

In this example, first you configure network interfaces on the device, create security zones and assign interfaces to the zones, and configure a policy to allow TALK traffic to go through an SRX Series Firewall.

Then you create a static NAT rule set rs1 with a rule r1 to match the destination address 40.5.2.120/32, and you create a static NAT prefix with address 20.5.2.120/32.

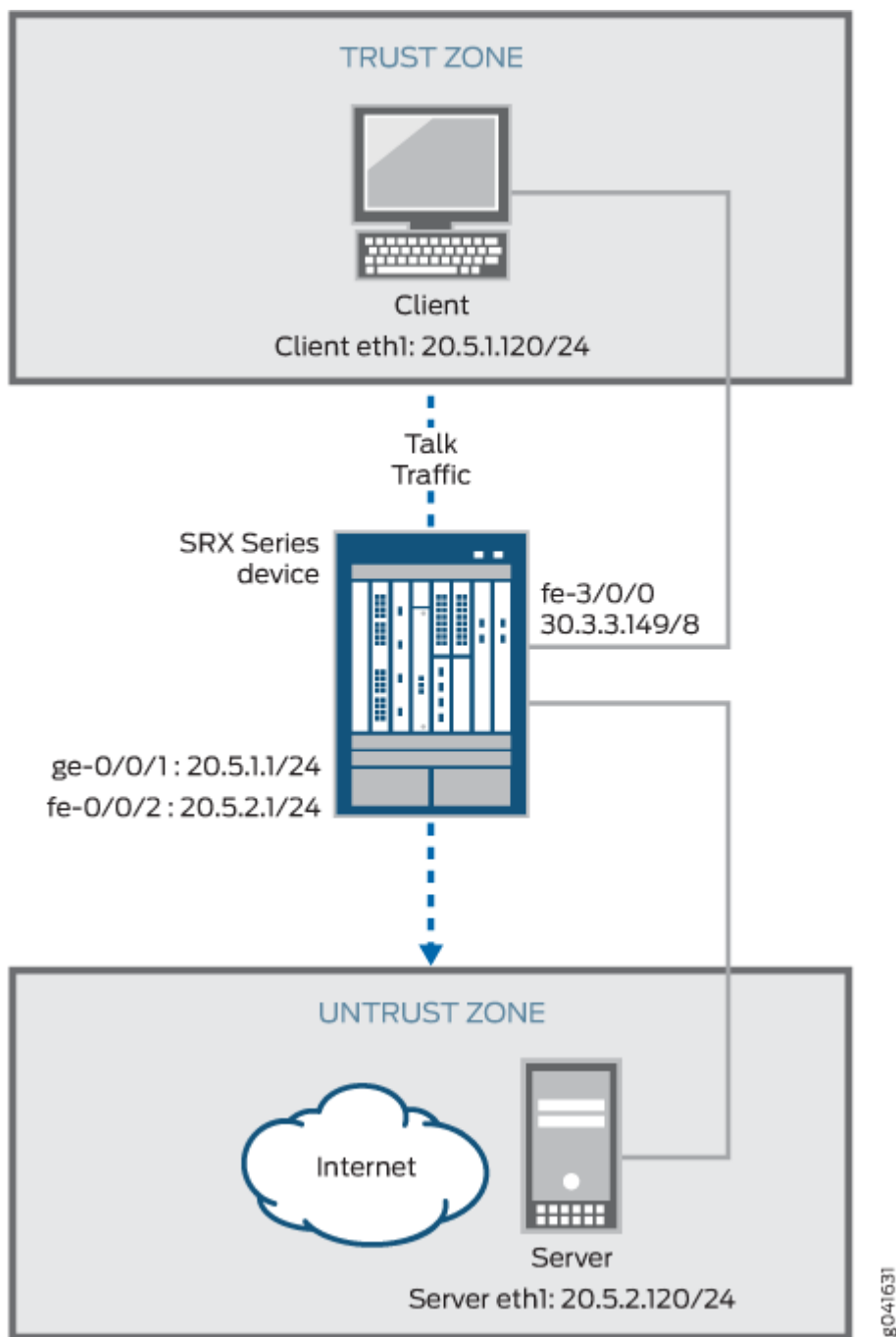
Next you create a source NAT pool src-p1 with a source rule set src-rs1 to translate packets from zone trust to zone untrust. For matching packets, the source address is translated to an IP address in the src-p1 pool.

Then you create a destination NAT pool des-p1 with a destination rule set des-rs1 to translate packets from zone trust to destination address 40.5.2.121/32. For matching packets, the destination address is translated to an IP address in the des-p1 pool. Finally, you configure TALK ALG trace options.

Topology

[Figure 10 on page 147](#) shows the TALK ALG topology.

Figure 10: TALK ALG Topology



Configuration

IN THIS SECTION

- [Configuring a Route Mode | 148](#)
- [Configuring a Static NAT Rule Set | 151](#)
- [Configuring a Source NAT Pool and Rule Set | 153](#)
- [Configuring a Destination NAT Pool and Rule Set | 155](#)
- [Configuring TALK ALG trace options | 157](#)

To configure the TALK ALG, perform these tasks:

Configuring a Route Mode

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family inet address 20.5.1.1/24
set interfaces fe-0/0/2 unit 0 family inet address 20.5.2.1/24
set security zones security-zone trust interfaces ge-0/0/1 host-inbound-traffic system-services
all
set security zones security-zone trust interfaces ge-0/0/1 host-inbound-traffic protocols all
set security zones security-zone untrust interfaces fe-0/0/2 host-inbound-traffic system-
services all
set security zones security-zone untrust interfaces fe-0/0/2 host-inbound-traffic protocols all
set security policies from-zone trust to-zone untrust policy talk match source-address any
set security policies from-zone trust to-zone untrust policy talk match destination-address any
set security policies from-zone trust to-zone untrust policy talk match application junos-ntalk
set security policies from-zone trust to-zone untrust policy talk then permit
```

Step-by-Step Procedure

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

To configure route mode:

1. Configure interfaces.

```
[edit interfaces]
user@host#set ge-0/0/1 unit 0 family inet address 20.5.1.1/24
user@host#set fe-0/0/2 unit 0 family inet address 20.5.2.1/24
```

2. Configure zones and assign interfaces to the zones.

```
[edit security zones security-zone trust]
user@host#set interfaces ge-0/0/1 host-inbound-traffic system-services all
user@host#set interfaces ge-0/0/1 host-inbound-traffic protocols all
[edit security zones security-zone untrust]
user@host#set interfaces fe-0/0/2 host-inbound-traffic system-services all
user@host#set interfaces fe-0/0/2 host-inbound-traffic protocols all
```

3. Configure a TALK policy that allows TALK traffic from the trust zone to the untrust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host#set policy talk match source-address any
user@host#set policy talk match destination-address any
user@host#set policy talk match application junos-ntalk
user@host#set policy talk then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

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

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

```

        family inet {
            address 20.5.1.1/24;
        }
    }
}
...
fe-0/0/2 {
    unit 0 {
        family inet {
            address 20.5.2.1/24;
        }
    }
}

```

```

[edit]
user@host# show security zones
security-zone trust {
    ....
    interfaces {
        ge-0/0/1.0 {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
        }
    }
}
...
security-zone untrust {
    interfaces {
        fe-0/0/2.0 {
            host-inbound-traffic {
                system-services {
                    all;
                }
                protocols {
                    all;
                }
            }
        }
    }
}

```

```

    }
  }
}

```

```

[edit]
user@host# show security policies
from-zone trust to-zone untrust {
  policy talk {
    match {
      source-address any;
      destination-address any;
      application junos-ntalk;
    }
    then {
      permit;
    }
  }
}

```

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

Configuring a Static NAT Rule Set

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```

set security nat static rule-set rs1 from zone trust
set security nat static rule-set rs1 rule r1 match destination-address 40.5.2.120/32
set security nat static rule-set rs1 rule r1 then static-nat prefix 20.5.2.120/32

```

Step-by-Step Procedure

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

To configure a static NAT rule set:

1. Create a static NAT rule set.

```
[edit security nat static rule-set rs1]
user@host# set from zone trust
```

2. Define the rule to match with the destination address.

```
[edit security nat static rule-set rs1]
user@host# set rule r1 match destination-address 40.5.2.120/32
```

3. Define the static NAT prefix for the device.

```
[edit security nat static rule-set rs1]
user@host# set rule r1 then static-nat prefix 20.5.2.120/32
```

Results

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

```
[edit]
user@host# show security nat
static {
  rule-set rs1 {
    from zone trust;
    rule r1 {
      match {
        destination-address 40.5.2.120/32
      }
      then {
        static-nat {
```


2. Create a source NAT rule set.

```
[edit security nat source]
user@host# set rule-set src-rs1 from zone trust
user@host# set rule-set src-rs1 to zone untrust
```

3. Configure a rule that matches packets and translates the source address to an address in the source pool.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule src-r1 match source-address 20.5.1.120/32
```

4. Configure a rule that matches packets and translates the destination address to an address in the source pool.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule src-r1 match destination-address 20.5.2.120/32
```

5. Configure a source NAT pool in the rule.

```
[edit security nat source]
user@host# set rule-set src-rs1 rule src-r1 then source-nat pool src-p1
```

Results

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

```
[edit]
user@host# show security nat
source {
  pool src-p1 {
    address {
      40.5.1.120/32;
    }
  }
}
rule-set src-rs1 {
```

```

from zone trust;
to zone untrust;
rule src-r1 {
    match {
        source-address 20.5.1.120/32;
        destination-address 20.5.2.120/32;
    }
    then {
        source-nat {
            pool {
                src-p1;
            }
        }
    }
}
}
}
}
}

```

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

Configuring a Destination NAT Pool and Rule Set

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```

set security nat destination pool des-p1 address 20.5.2.120/32
set security nat destination rule-set des-rs1 from zone trust
set security nat destination rule-set des-rs1 rule des-r1 match source-address 20.5.1.120/32
set security nat destination rule-set des-rs1 rule des-r1 match destination-address 40.5.2.120/32
set security nat destination rule-set des-rs1 rule des-r1 then destination-nat pool des-p1

```

Step-by-Step Procedure

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

To configure a destination NAT pool and rule set:

1. Create a destination NAT pool.

```
[edit security nat destination]
user@host# set pool des-p1 address 20.5.2.120/32
```

2. Create a destination NAT rule set.

```
[edit security nat destination]
user@host# set rule-set des-rs1 from zone trust
```

3. Configure a rule that matches packets and translates the source address to the address in the pool.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 match source-address 20.5.1.120/32
```

4. Configure a rule that matches packets and translates the destination address to the address in the pool.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 match destination-address 40.5.2.120/32
```

5. Configure a source NAT pool in the rule.

```
[edit security nat destination]
user@host# set rule-set des-rs1 rule des-r1 then destination-nat pool des-p1
```

Results

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

```
[edit]
user@host# show security nat
destination {
  pool des-p1 {
    address {
```

```

        20.5.2.120/32;
    }
}
rule-set des-rs1 {
    from zone trust;
    rule des-r1 {
        match {
            source-address 20.5.1.120/32;
            destination-address 40.5.2.120/32;
        }
        then {
            destination-nat {
                pool {
                    des-p1;
                }
            }
        }
    }
}
}

```

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

Configuring TALK ALG trace options

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```

set security alg talk traceoptions flag all
set security alg traceoptions file trace
set security alg traceoptions file size 1g
set security alg traceoptions level verbose

```

Step-by-Step Procedure

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

To configure TALK ALG trace options:

1. Enable TALK ALG trace options.

```
[edit security alg]
user@host#set talk traceoptions flag all
```

2. Configure the filename to receive output from the tracing operation.

```
[edit security alg]
user@host#set traceoptions file trace
```

3. Specify the maximum trace file size.

```
[edit security alg]
user@host#set traceoptions file size 1g
```

4. Specify the level of tracing output.

```
[edit security alg]
user@host#set traceoptions level verbose
```

Results

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

```
[edit]
user@host# show security alg
traceoptions {
    file trace size 1g;
    level verbose;
}
talk traceoptions flag all;
```

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

Verification

IN THIS SECTION

- [Verifying the TALK ALG Control Session | 159](#)
- [Verifying the TALK Flow Gate Information | 160](#)
- [Verifying TALK ALG | 161](#)
- [Verifying the TALK Resource Manager Group | 162](#)
- [Verifying the TALK Resource Information | 162](#)

Confirm that the configuration is working properly.

Verifying the TALK ALG Control Session

Purpose

Verify that the TALK control session is created and all the TALK control and data sessions are created.

Action

From operational mode, enter the `show security flow session` command.

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

```

    session ID: 128570, Policy name: p11/4, Timeout: 56, Valid
Resource information : TALK ALG, 2, 0
  In: 5.1.1.200/1105 --> 6.1.1.200/518;udp, If: ge-0/0/1.0, Pkts: 3, Bytes: 336
  Out: 6.1.1.200/518 --> 5.1.1.200/1105;udp, If: ge-0/0/2.0, Pkts: 3, Bytes: 156

Session ID: 128617, Policy name: p11/4, Timeout: 1796, Valid
Resource information : TALK ALG, 2, 2
  In: 6.1.1.200/42224 --> 5.1.1.200/518;udp, If: ge-0/0/2.0, Pkts: 1, Bytes: 112
  Out: 5.1.1.200/518 --> 6.1.1.200/42224;udp, If: ge-0/0/1.0, Pkts: 1, Bytes: 52

Session ID: 128618, Policy name: p11/4, Timeout: 1796, Valid
Resource information : TALK ALG, 2, 3
```

```
In: 6.1.1.200/51430 --> 5.1.1.200/32905;tcp, If: ge-0/0/2.0, Pkts: 4, Bytes: 219
Out: 5.1.1.200/32905 --> 6.1.1.200/51430;tcp, If: ge-0/0/1.0, Pkts: 3, Bytes: 167
```

Meaning

- **Session ID**—Number that identifies the session. Use this ID to get more information about the session such as policy name or number of packets in and out.
- **Policy name**—Policy name that permitted the traffic.
- **In**—Incoming flow (source and destination IP addresses with their respective source and destination port numbers, session is TCP, and the source interface for this session is ge-0/0/1.0).
- **Out**—Reverse flow (source and destination IP addresses with their respective source and destination port numbers, session is TCP, and destination interface for this session is fe-0/0/2.0).

Verifying the TALK Flow Gate Information

Purpose

Verify that the gates are opened for TCP data channel and reverse UDP reply.

Action

From operational mode, enter the `show security flow gate` command.

```
user@host>show security flow gate
Hole: 6.1.1.200-6.1.1.200/0-0->5.1.1.200-5.1.1.200/518-518
  Translated: 0.0.0.0/0->0.0.0.0/0
  Protocol: udp
  Application: TALK ALG/65
  Age: 110 seconds
  Flags: 0x0080
  Zone: untrust
  Reference count: 1
  Resource: 11-2-2

Hole: 6.1.1.200-6.1.1.200/0-0->5.1.1.200-5.1.1.200/32905-32905
  Translated: 0.0.0.0/0->0.0.0.0/0
  Protocol: tcp
  Application: TALK ALG/65
  Age: 110 seconds
```

```

Flags: 0x0080
Zone: untrust
Reference count: 1
Resource: 11-2-3

```

Meaning

- **Hole**—Range of flows permitted by the pinhole.
- **Translated**—Tuples used to create the session if it matches the pinhole (source and destination IP addresses with their respective source and destination port numbers).
- **Protocol**—Application protocol, such as UDP or TCP.
- **Application**—Name of the application.
- **Age**—Idle timeout for the pinhole.
- **Flags**— Internal debug flags for the pinhole.
- **Zone**—Security zone such as from zone and to zone.
- **Reference count**—Number of resource manager references to the pinhole.
- **Resource**—Resource manager information about the pinhole.

Verifying TALK ALG

Purpose

Verify that the TALK ALG is enabled.

Action

From operational mode, enter the `show security alg status` command.

```

user@host>show security alg status
ALG Status :
  PPTP      : Enabled
  RSH       : Disabled
  RTSP      : Enabled
  SCCP      : Enabled
  SIP       : Enabled

```

```
TALK      : Enabled
TFTP      : Enabled
IKE-ESP   : Disabled
```

Meaning

The output shows the TALK ALG status as follows:

- Enabled—Shows the TALK ALG is enabled.
- Disabled—Shows the TALK ALG is disabled.

Verifying the TALK Resource Manager Group

Purpose

Verify the total number of resource manager groups and active groups that are used by the TALK ALG.

Action

From operational mode, enter the `show security resource-manager group active` command.

```
user@host>show security resource-manager group active
Group ID 2: Application - TALK ALG
Total groups 3276, active groups 1
```

Verifying the TALK Resource Information

Purpose

Verify the total number of resources and active resources that are used by the TALK ALG.

Action

From operational mode, enter the `show security resource-manager resource active` command.

```
user@host>show security resource-manager resource active
Resource ID 3: Group ID - 2, Application - TALK ALG
```

Resource ID 2: Group ID - 2, Application - TALK ALG
Total Resources 6015, active resources 2

RELATED DOCUMENTATION

[IKE and ESP ALG | 37](#)

[PPTP ALG | 53](#)

TFTP ALG

IN THIS SECTION

- [Understanding the TFTP ALG | 164](#)
- [Example: Configuring the TFTP ALG | 166](#)
- [Platform-Specific TFTP ALG Behavior | 171](#)

The Trivial File Transfer Protocol (TFTP) ALG processes TFTP packets that initiate the request to UDP destination port 69 and opens a gate to allow return packets from the reverse direction to the port that sends the request. Support of stateful firewall and NAT services requires that you configure the TFTP ALG for UDP destination port 69.

Use [Feature Explorer](#) to confirm platform and release support for specific features.

Review the "[Platform-Specific TFTP ALG Behavior](#)" on [page 171](#) section for notes related to your platform.

Understanding the TFTP ALG

IN THIS SECTION

- [Overview | 164](#)
- [TFTP Packets | 164](#)
- [TFTP Session | 165](#)
- [Understanding TFTP ALG Conversation | 165](#)
- [Understanding IPv6 Support for the TFTP ALG | 166](#)

Overview

Trivial File Transfer Protocol (TFTP) is a simple protocol used for files transfer (*RFC 1350*). TFTP is implemented on top of UDP, with destination port 69 as the well-known port. The TFTP Application Layer Gateway (ALG) processes TFTP packets that initiate the request and creates pinholes to allow return packets from the reverse direction.

In flow processing there are two sessions for one TFTP conversation, one is the TFTP control session created by a read request (RRQ) or write request (WRQ) packet; the other one is the TFTP data session created by a DATA packet (for RRQ) or acknowledgment (ACK) packet (for WRQ).

In a Junos OS firewall, the TFTP control session is permitted through the `junos-tftp` application policy. The data session is permitted through the TFTP ALG open pinhole from any port of the server to the TID (port) of the client when the control session packet is received. No NAT translation is required, because the NAT translation has already been performed and the information is available from the session data structure.

TFTP Packets

Any transfer begins with a request to read or write a file. A data packet of less than 512 bytes signals termination of a transfer.

TFTP supports five types of packets:

- Read request (RRQ)
- Write request (WRQ)
- Data (DATA)

- Acknowledgment (ACK)
- Error (ERROR)

TFTP Session

The TFTP ALG is based on UDP, which is a stateless transport protocol. In a firewall, the TFTP ALG acts as a UDP session with timeout. If there is no packet refresh session, the session is terminated after timeout. Although the TFTP client and server determine the termination of a TFTP conversation, they are sometimes unaware of the session in Firewall. Therefore, the client and server could request a new TFTP conversation in this scenario.

The TFTP ALG session can proceed in any of the following ways:

- When the TFTP control session reaches timeout, the session is not terminated if the data session is still alive.
- A TFTP session might terminate or get corrupted by the clear security flow session all or the clear specific session CLI commands regardless of whether the data session is ongoing or not.
- If a new TFTP session request arrives and reaches the existing session, the TFTP ALG will open the pinhole again for the new request.
- If the pinhole already exists, the TFTP ALG will not open the pinhole again and there will be no packet drop.
- The TFTP ALG will not drop any packet.

Understanding TFTP ALG Conversation

By default TFTP servers listen for incoming requests from TFTP clients on port 69. A TFTP client chooses its source tunnel identifier (TID) port and sends its initial request to the server. In response, the server uses the TID chosen as the source port and sends a response to the client's TID as the destination port. The two TIDs ports are then used for the rest of the data transfer.

Read file conversation steps:

1. Host A (client) sends an RRQ packet to host B (server) with A's TID as source and port 69 as destination.
2. Host B (server) sends a DATA packet to host A (client) with B's TID as source and A's TID as destination.
3. Host A (client) sends an ACK packet to host B (server) with A's TID as source and B's TID as destination.

4. DATA and ACK packets conversation continues until file data transferring is complete.

Write file conversation steps:

1. Host A (client) sends a WRQ packet to host B (server) with A's TID as source and port 69 as destination.
2. Host B (server) sends an ACK packet to host A (client) with B's TID as source and A's TID as destination.
3. Host A (client) sends a DATA packet to host B (server) with A's TID as source and B's TID as destination.
4. Host B (server) sends an ACK packet to host A (client) with B's TID as source and A's TID as destination.

Understanding IPv6 Support for the TFTP ALG

Trivial File Transfer Protocol (TFTP) Application Layer Gateway (ALG) has been enhanced to support IPv6 and IPv4 TFTP conversation, which has IPv6 and IPv4 addresses for both the source IP address and destination IP address.

TFTP ALG processes packets that initiate the routing request and create pinholes to allow return packets from the reverse direction to the port that sent the request.

The data session is set up by the first packet from the client to the server. TFTP ALG monitors the first packet and opens a pinhole from any port on the server to the client. This process helps the return packets from the server and subsequent data packets to pass through.

Example: Configuring the TFTP ALG

IN THIS SECTION

- [Requirements | 167](#)
- [Overview | 167](#)
- [Configuration | 167](#)
- [Verification | 170](#)

The TFTP ALG processes TFTP packets that initiate the request and opens a gate to allow return packets from the reverse direction to the port that sends the request.

This example shows how to configure the TFTP ALG to pass through TFTP traffic with a source NAT pool on Juniper Networks devices.

Requirements

- Configure proxy ARP for all IP addresses in the source NAT pool.
- Understand the basic concepts of TFTP ALG. See ["Understanding the TFTP ALG" on page 164](#).

Overview

In this example, the TFTP ALG is configured to monitor and allow TFTP traffic, transferring files between the client and server located on opposite sides of a Juniper Networks device.

Configuration

IN THIS SECTION

- [Configuring a NAT Source Pool, Rule Set, and a Policy | 167](#)

Configuring a NAT Source Pool, Rule Set, and a Policy

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security nat source pool pool1 address 10.10.10.1/32 to 10.10.10.10/32
set security zones security-zone green address-book address sa1 1.1.1.0/24
set security zones security-zone red address-book address da1 2.2.2.0/24
set security nat source rule-set rs1 from zone green
set security nat source rule-set rs1 to zone red
set security nat source rule-set rs1 rule r1 match source-address 1.1.1.0/24
```

```
set security nat source rule-set rs1 rule r1 match destination-address 2.2.2.0/24
set security nat source rule-set rs1 rule r1 then source-nat pool pool1
```

```
set security policy from-zone green to-zone red policy pol1 match destination-address da1
set security policy from-zone green to-zone red policy pol1 match source-address sa1
set security policy from-zone green to-zone red policy pol1 match application junos-tftp
set security policy from-zone green to-zone red policy pol1 then permit
```



NOTE: If you are not sure of the TFTP client and server IP address, you can replace “da1” and “sa1” with “any”.

Step-by-Step Procedure

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

To configure a source NAT pool:

1. Create a NAT source pool.

```
[edit security]
user@host# set nat source pool pool1 address 10.10.10.1/32 to 10.10.10.10/32
```

2. Configure security zone address book entries.

```
[edit security zones security-zone]
user@host# set green address-book address sa1 1.1.1.0/24
user@host# set red address-book address da1 2.2.2.0/24
```

3. Create a NAT source rule set.

```
[edit security nat source rule-set rs1]
user@host# set from zone green
user@host# set to zone red
user@host# set rule r1 match source-address 1.1.1.0/24
user@host# set rule r1 match destination-address 2.2.2.0/24
user@host# set rule r1 then source-nat pool pool1
```

4. Configure a policy

```
[edit security policies from-zone green to-zone red policy pol1]
user@host# set match source-address sa1
user@host# set match destination-address da1
user@host# set match application junos-tftp
user@host# set then permit
```

Results

From configuration mode, confirm your configuration by entering the `show security nat` and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show security nat
source {
    pool pool1 {
        address {
            10.10.10.1/32 to 10.10.10.10/32;
        }
    }
}
rule-set rs1 {
    from zone green;
    to zone red;
    rule r1 {
        match {
            source-address 1.1.1.0/24;
            destination-address 2.2.2.0/24;
        }
        then {
            source-nat {
                pool {
                    pool1;
                }
            }
        }
    }
}
```

```
    }
}
```

```
[edit]
user@host# show security policies
from-zone green to-zone red {policy pol1 {
  policy pol1 {
    match {
      source-address sa1;
      destination-address da1;
      application [junos-tftp];
    }
    then {
      permit;
    }
  }
}
default-policy {
  permit-all;
}
```

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

Verification

IN THIS SECTION

- [Verifying the NAT Source Pool and Rule Set | 170](#)
- [Verifying TFTP ALG | 171](#)

Confirm that the configuration is working properly.

Verifying the NAT Source Pool and Rule Set

Purpose

Verify that the NAT source pool and rule set used to support the TFTP ALG are working properly.

Action

From operational mode, enter the `show security nat static rule r1` command.

Verifying TFTP ALG

Purpose

Verify that the TFTP ALG is enabled.

Action

From operational mode, enter the `show security alg status` command.

```
user@host> show security alg status
```

```
DNS      : Enabled
FTP       : Enabled
H323     : Enabled
TFTP     : Enabled
```

Meaning

The output shows the TFTP ALG status as follows:

- Enabled—Shows the TFTP ALG is enabled.
- Disabled—Shows the TFTP ALG is disabled.

Platform-Specific TFTP ALG Behavior

Use [Feature Explorer](#) to confirm platform and release support for specific features.

Use the following table to review platform-specific behaviors for your platform:

Platform	Difference
SRX Series	<ul style="list-style-type: none"> SRX320, SRX340, and SRX380 devices that support TFTP do not support broadcast TFTP when flow is enabled on the device.

RELATED DOCUMENTATION

[RPC ALG | 72](#)

[RSH ALG | 87](#)

TWAMP ALG

IN THIS SECTION

- [Understanding the Two-Way Active Measurement Protocol \(TWAMP\) Application Layer Gateway \(ALG\) | 173](#)
- [Enabling the Two-Way Active Measurement Protocol \(TWAMP\) Application Layer Gateway \(ALG\) | 175](#)

The Two-Way Active Measurement Protocol (TWAMP) is an open protocol for measuring network performance between any two devices in a network that supports the protocols in the TWAMP framework. The TWAMP Application Layer Gateway (ALG) extracts the IP addresses and the port details from messages between the control client and the TWAMP server and performs the IP address translation and gate opening to permit test sessions.

Understanding the Two-Way Active Measurement Protocol (TWAMP) Application Layer Gateway (ALG)

IN THIS SECTION

- [Understanding TWAMP ALG | 173](#)
- [Limitations for TWAMP ALG | 174](#)

The Two-Way Active Measurement Protocol (TWAMP) is an open protocol for measuring network performance between any two devices in a network that supports the protocols in the TWAMP framework.

Starting from Junos OS Release 18.2R1, the TWAMP ALG is supported to enable the TWAMP data traffic to pass through the NFX Series or SRX Series Firewall without needing a predefined policy permission.

An Application Layer Gateway (ALG) is a software component that is designed to manage the specific protocols such as Session Initiation Protocol (SIP) or FTP on Juniper Networks devices running Junos OS. The ALG module is responsible for application layer aware packet processing.

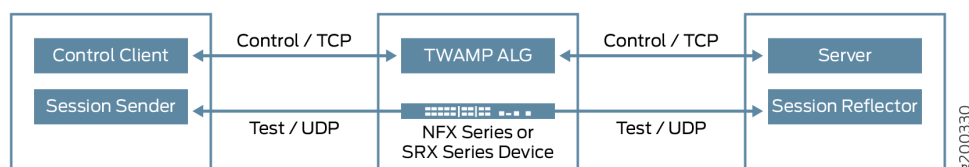
The TWAMP ALG is located between the control client and the TWAMP server. The TWAMP ALG extracts the IP addresses and the port details from messages between the control client and the TWAMP server and performs the IP address translation and gate opening to permit test sessions. The TWAMP ALG performs the following functions:

- Decoding all the TWAMP control messages.
- Performing the TWAMP message sanity checks and decoding.
- Performing the IP address translation when the device is configured with NAT.

Understanding TWAMP ALG

The TWAMP consists of the following inter-related protocols:

- The TWAMP-Control is used to initiate, start and stop the test sessions between the control client and the TWAMP server.
- The TWAMP-Test is used to exchange the test packets between the session sender and the session reflector.

Figure 11: TWAMP ALG

As shown in [Figure 11 on page 174](#), the control client initiates all requested test sessions with the TWAMP server through control connection. Meantime, the TWAMP ALG extracts the IP addresses and the port details in the messages from both sides to open a pinhole to permit the test sessions. During this stage, the TWAMP ALG might bypass the packets if the TWAMP server and the TWAMP client support unauthenticated mode.

In this case, the session sender and the reflector can exchange the test packets according to the TWAMP-Test protocol for each active test session.

Limitations for TWAMP ALG

The following are the limitations for the TWAMP ALG:

- For the TWAMP connections, the TWAMP client and the TWAMP server must re-initiate a new control connection for test in case of failover.
- During the control connection negotiation, if the servers rejects the client request, the client or the server must close the connection.
- The TWAMP ALG has a timeout period of four seconds to establish the session.
- The TWAMP ALG requires that the sender address, receiver address, and the header destination IP from the same interface to translate the IP address in the payload.

SEE ALSO

Two-Way Active Measurement Protocol (TWAMP) Overview

Enabling the Two-Way Active Measurement Protocol (TWAMP) Application Layer Gateway (ALG)

The Application Layer Gateway (ALG) Two-Way Active Measurement Protocol (TWAMP) is supported to enable the TWAMP data traffic to pass through the NFX Series or SRX Series Firewall without needing a predefined policy permission. By default, the ALG TWAMP is disabled.

To enable the TWAMP ALG and traceoptions:

1. Enable the TWAMP ALG.

```
[edit security]
user@host# alg twamp
```

2. Enable the TWAMP traceoptions.

```
[edit security]
user@host# alg twamp traceoptions flag all
```

Use the `show security alg status` command to verify the status of the TWAMP ALG.

```
user@host> show security alg status
DNS      : Enabled
FTP      : Enabled
H323     : Disabled
MGCP     : Disabled
MSRPC    : Enabled
PPTP     : Enabled
RSH      : Disabled
RTSP     : Disabled
SCCP     : Disabled
SIP      : Disabled
SQL      : Disabled
SUNRPC   : Enabled
TALK     : Enabled
TFTP     : Enabled
IKE-ESP  : Disabled
TWAMP    : Disabled
```

SEE ALSO

| [twamp \(Security ALG\)](#)

Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
18.2R1	Starting from Junos OS Release 18.2R1, the TWAMP ALG is supported to enable the TWAMP data traffic to pass through the NFX Series or SRX Series Firewall without needing a predefined policy permission.

RELATED DOCUMENTATION

| [Understanding Data ALG Types](#) | 13

Understanding IPv6 ALG Support for ICMP

IN THIS SECTION

- [ICMP Error Messages](#) | 177
- [ICMP ALG Functionality](#) | 177

The Internet Control Message Protocol (ICMP) Application Layer Gateway (ALG) is one of the ALG's that handle ICMP traffic.

IPv6 nodes use the ICMPv6 protocol to report errors encountered in processing packets and to perform other Internet-layer functions such as diagnostics. ICMPv6 is an integral part of IPv6 and must be fully implemented by every IPv6 node; therefore the ALG layer is always enabled for ICMPv6.

ICMP Error Messages

ICMPv6 messages are grouped into two classes:

- ICMPv6 error messages
 - Destination unreachable
 - Packet too big
 - Time exceeded
 - Parameter problem
- ICMPv6 informational (or ping) messages
 - Echo request
 - Echo reply

The ICMP ALG monitors all these messages, and then does the following :

- Closes the session
- Modifies the payload

The ICMP ALG closes a session if it meets the following conditions:

- Receives echo reply message.
- Receives a destination unreachable error message and has not received any replies yet.



NOTE: The ICMP ALG checks if the session has received any replies from destination node. If it has received any reply , the destination should be reachable and the ICMP error message is not credible, therefore it does not close the session. This is to avoid hackers from sniffing the TCP/UDP packet and forging an ICMP destination unreachable packet to kill the session.

ICMP ALG Functionality

ICMP ALG behaves differently in various modes.

ICMP ALG functionality in NAT mode:

1. Close the session.
2. Modify the identifier, the sequence number or both of the echo request.
3. Resume the original identifier and sequence number for the echo reply.
4. NAT translates the embedded IPv6 packet for the ICMPv6 error message.

ICMP ALG functionality in NAT-PT support mode:

1. Close the session.
2. Translate the ICMPv4 ping message to the ICMPv6 ping message.
3. Translate the ICMPv6 ping message to the ICMPv4 ping message.
4. Translate the ICMPv4 error message to the ICMPv6 error message and translate its embedded IPv4 packet to an IPv6 packet.
5. Translate the ICMPv6 error message to the ICMPv4 error message and translate its embedded IPv6 packet to an IPv4 packet .

RELATED DOCUMENTATION

[Understanding How SRX Series Devices Handle Packet Fragmentation for IPv6 Flows](#)

[Understanding IPv6 Address Space, Addressing, Address Format, and Address Types](#)

Understanding 464XLAT ALG Traffic Support

IN THIS SECTION

- [Understanding 464XLAT ALG Functionality | 180](#)
- [How the PPTP ALG Supports the Device Acting As PLAT | 181](#)
- [How the RTSP ALG Supports the Device Acting As PLAT | 182](#)
- [How the FTP ALG Supports the Device Acting As PLAT | 184](#)

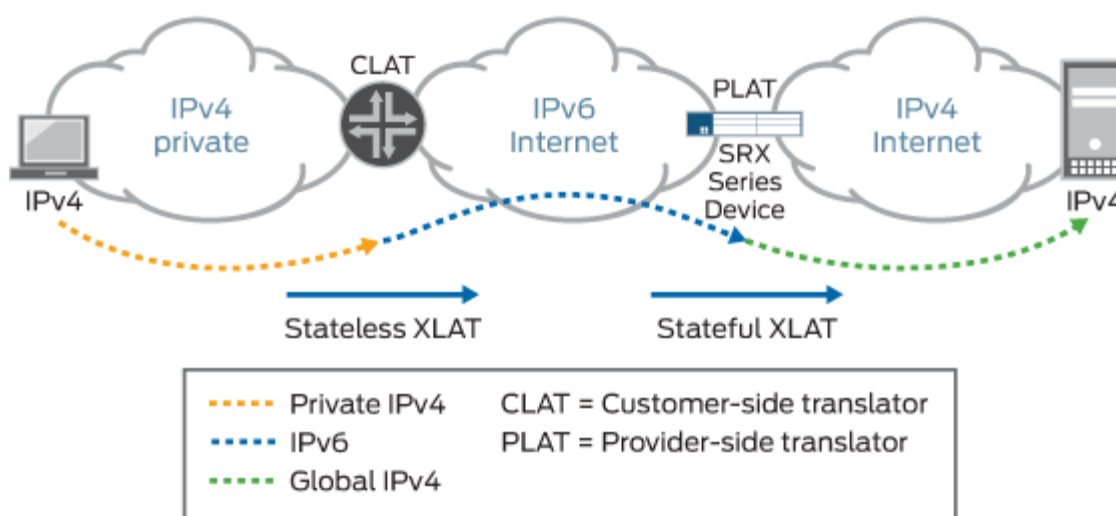
When you deploy IPv6 applications on mobile networks, be aware that some mobile operators cannot provide IPv6 support for their users, because some phone applications do not support an IPv6-only environment.

The solution is to use the NAT64 mechanism to access the IPv4-only content in the operator's network and to use *464XLAT* traffic to enable IPv4-only applications to work on IPv6-only networks.

The 464XLAT architecture is a combination of stateless translation on the customer-side translator (CLAT) and stateful translation on the provider-side translator (PLAT). The 464XLAT architecture is used to translate the packet information of a device using the combination of stateless (translates private IPv4 address to global IPv6 addresses, and vice versa) and stateful (translates IPv6 addresses to global IPv4 addresses, and vice versa) translation.

Figure 12 on page 179 illustrates the 464XLAT architecture, which provides IPv4 connectivity across an IPv6-only network by combining existing and well-known stateful protocol translation on PLAT in the core and stateless protocol on CLAT at the edge. The private IPv4 host can reach global IPv4 hosts through both CLAT and PLAT translation. Conversely, the IPv6 host can directly reach other IPv6 hosts on the Internet without translation. This means that the customer premises equipment (CPE) can support CLAT and also operate as an IPv6 native router for native IPv6 traffic.

Figure 12: 464XLAT Architecture

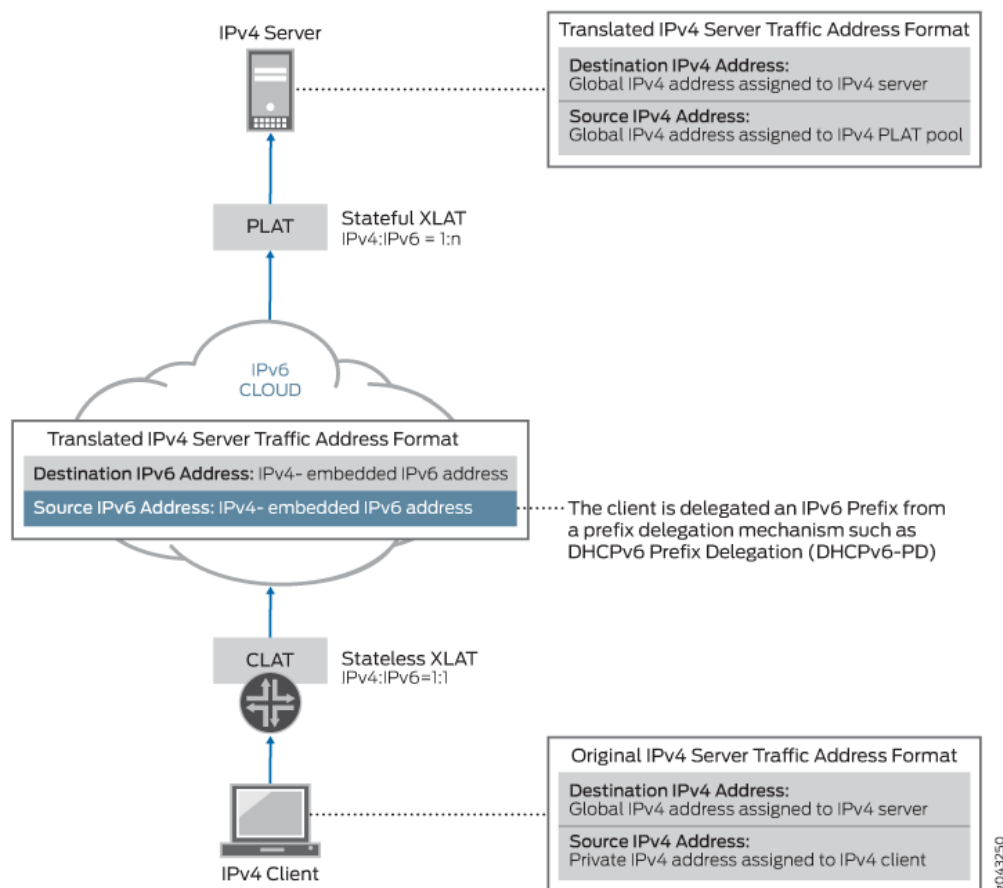


8043061

Understanding 464XLAT ALG Functionality

Figure 13 on page 180 describes the address translation architecture and shows how packet information for a device is translated using a combination of stateful translation at the provider-side translator (PLAT) and stateless translation at the customer-side translator (CLAT). In this diagram, the client is delegated an IPv6 prefix from a prefix delegation mechanism such as DHCPv6 Prefix Delegation (DHCPv6-PD). Therefore, the client has a dedicated IPv6 prefix for translation.

Figure 13: 464XLAT ALG Functionality



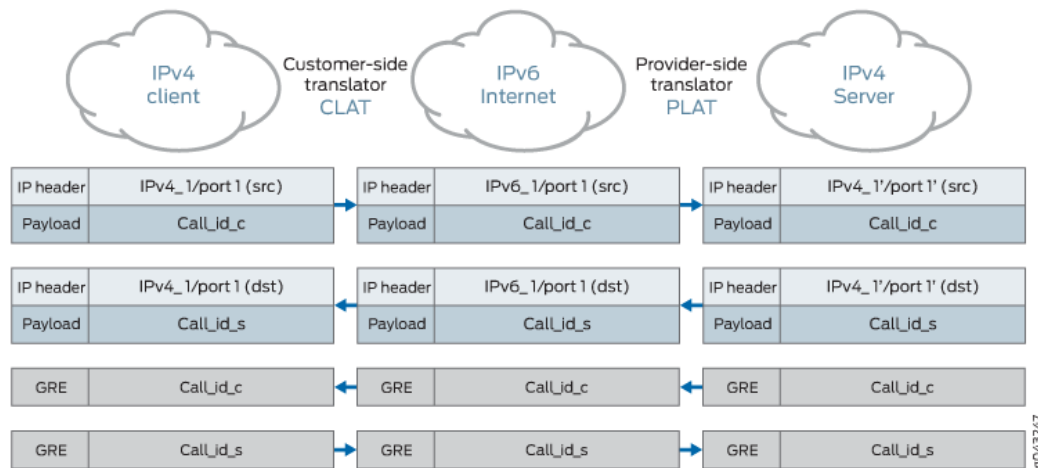
The PPTP, RTSP, and FTP ALGs also support XLAT functionality.

This following sections explain how the PPTP, RTSP, and FTP ALGs work when the device acts as PLAT:

How the PPTP ALG Supports the Device Acting As PLAT

Figure 14 on page 181 describes the PPTP ALG XLAT functionality.

Figure 14: PPTP ALG XLAT Functionality



The PPTP ALG uses the call_ID for destination port functionality.

1. The client sends the outgoing call request (with PPTP Access Concentrator (PAC) call_ID) to the server:

CLAT: The source address/port is translated from lpv4_1/port1 to lpv6_1/port1. However, the payload call_ID is not changed.

PLAT: The source address/port lpv6_1/port1 is translated to lpv4_1'/port1' and matches the NAT64 rule. However, the call_ID in the payload is not changed. The PPTP ALG creates a gate such as server_ip/0->lpv4_1'/call_ID(lpv6_1/call_ID).

The first generic routing encapsulation (GRE) packet reaches the gate from the server side: When the first GRE traffic reaches the gate, the GRE packet from the server side with destination lpv4_1'/call_ID is translated to lpv6_1/call_ID. Finally, the GRE packet reaches the client lpv4_1/call_ID after CLAT.

Another special case for call_ID 0:

CLAT: The source address/port is translated from lpv4_1/port1 to lpv6_1/port1. However, the payload call_ID is not changed.

PLAT: The source address/port Ipv6_1/port1 is translated to Ipv4_1'/port1' and matches the NAT64 rule. However, the call_ID 0 in the payload is manually translated to 65002. The PPTP ALG creates a gate such as server_ip/0->Ipv4_1'/65002(Ipv6_1/0).

The first GRE packet reaches the gate from the server side: When the first GRE traffic reaches the gate, the GRE packet from the server side with destination Ipv4_1'/65002 is translated to Ipv6_1/0. Finally, the GRE packet reaches the client Ipv4_1/0 after CLAT.

2. The server sends the outgoing call reply (with PPTP Network Server (PNS) and PAC call_ID) to the client:

PLAT: The source address/port Ipv4_2/port2 is translated to Ipv6_2/port2' and matches the NAT64 rule. However, the call_ID in the payload is not changed, and the PPTP ALG creates a gate such as client_v6/0->Ipv6_2/call_ID(Ipv4_2/call_ID).

CLAT: The source address/port is translated from Ipv6_2/port2 to Ipv4_2/port2. However, the payload call_ID is not changed.

The first GRE packet reaches the gate from the client side: When the first GRE traffic reaches the gate, the GRE packet from the client side with destination Ipv4_2'/call_ID is translated to Ipv6_2/call_ID after CLAT and then it is translated to Ipv4_2/call_ID. Finally, the GRE packet reaches the server Ipv4_2/call_ID after PLAT.

Another special case for call_ID 0:

PLAT: The source address/port Ipv4_2/port2 is translated to Ipv6_2/port2' and matches the NAT64 rule. However, the call_ID in the payload is translated to 65002 and the PPTP ALG creates a gate such as client_v6/0->Ipv6_2/65002(Ipv4_2/0).

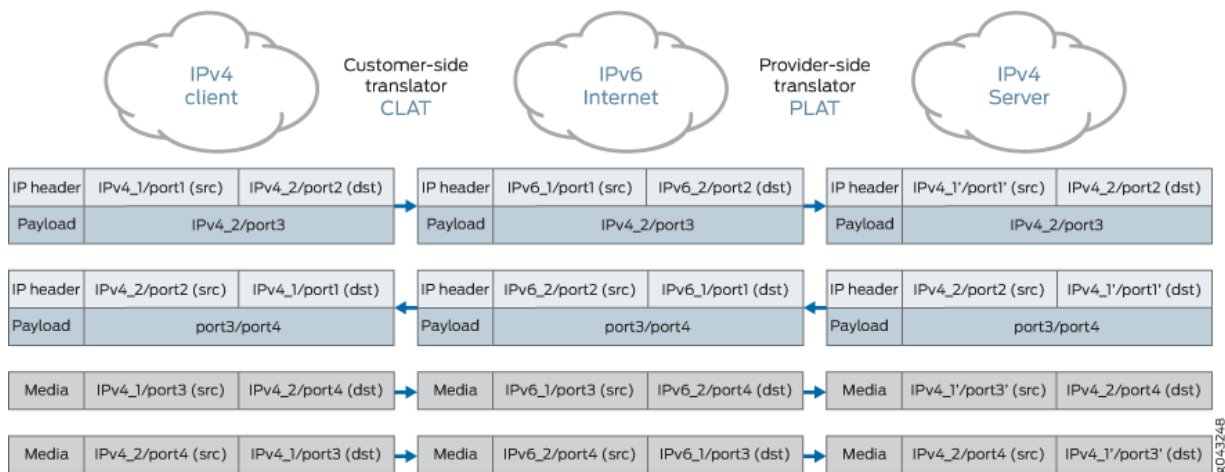
CLAT: The source address/port is translated from Ipv6_2/port2 to Ipv4_2/port2. However, the payload call_ID is not changed.

The first GRE packet reaches the gate from the client side: When the first GRE traffic reaches the gate, the GRE packet from the client side with destination Ipv4_2'/65002 is translated to Ipv6_2/65002 after CLAT and then it is translated to Ipv4_2/0. Finally, the GRE packet reaches the server Ipv4_2/0 after PLAT.

How the RTSP ALG Supports the Device Acting As PLAT

Figure 15 on page 183 describes the RTSP ALG XLAT functionality.

Figure 15: RTSP ALG XLAT Functionality



1. The Windows Media Player on the Windows PC sends a SETUP message:

CLAT: The source address/port is translated from lpv4_1/port1 to lpv6_1/port1. However, the payload lpv4_2/port3 is not changed.

PLAT: The source address/port lpv6_1/port1 is translated to lpv4_1'/port1' and matches the NAT64 rule, and the payload port3 is translated to port3'. However, the IP address in the payload ULR remains unchanged.

2. The Windows Media Server on the Windows server sends a 200 OK message:

PLAT: The source address/port lpv4_1'/port1' is translated to lpv6_1/port1 and matches the NAT64 rule. However, the port4 in the payload is not changed. The port3' is translated to port3. The RTSP ALG create gates such as c->s lpv6_1/port1->lpv6_2/port3 and s->c lpv4_2/port4->lpv4_1'/port3' over UDP media data sent from the server side with destination lpv4_1'/port1', then the IP header is translated to lpv6_1/port1 and reaches the gate.

CLAT: The source address/port is translated from lpv6_1/port1 to lpv4_1/port1. However, the payload port3/port4 is not changed.

3. The server sends the Real-Time Transport Protocol (RTP) over UDP media data:

PLAT: When the RTP over UDP media data is sent from the server side with destination lpv4_1'/port3, the IP header is translated to lpv6_1/port3 and reaches the gate.

CLAT: The IP header is translated from lpv6_1/port3 to lpv4_1/port3.

4. The client sends the RTP over UDP media data:

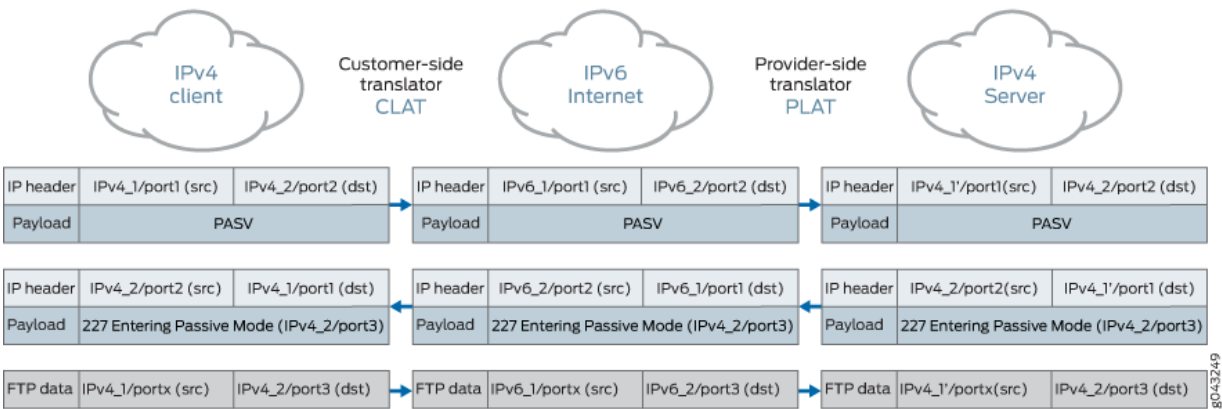
CLAT: The source address/port is translated from lpv4_1/port3 to lpv6_1/port3 and the destination address is translated from lpv4_2/port4 to lpv6_2/port4.

PLAT: The source address/port is translated from lpv6_1/port3 to lpv4_1'/port3 and the destination address is translated from lpv6_2/port4 to lpv4_2/port4.

How the FTP ALG Supports the Device Acting As PLAT

Figure 16 on page 184 and Figure 17 on page 185 describe the FTP ALG XLAT functionality in passive mode and port mode.

Figure 16: FTP Passive mode:



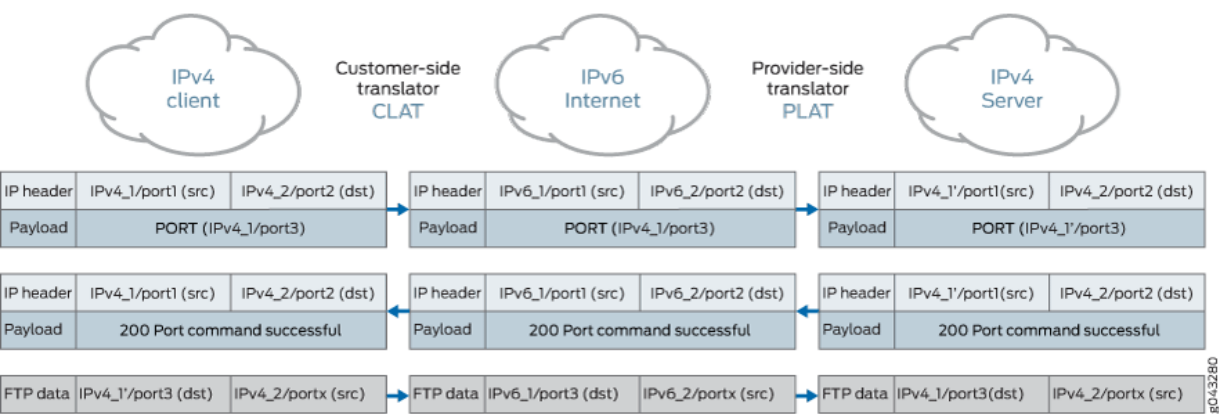
1. A 227 message enters passive mode:

CLAT: The source address/port is translated from lpv4_1/port1 to lpv6_1/port1. However, the payload does not contain IP or port information.

PLAT: The source address/port lpv4_1'/port1' is translated to lpv6_1/port1 and matches the NAT64 rule. However, the lpv4_2/port3 in the payload is not changed, and the FTP ALG creates a gate such as lpv4_1/0(lpv6_1/0)->lpv4_2/port3.

2. The first packet reaches the gate from the client side: When the traffic reaches the gate, the date packet from the client side with destination lpv4_2/port3 is translated to lpv6_2/port2. The IP header is translated to lpv4_2/port3 by NAT64 rule based on PLAT.

Figure 17: FTP Port Mode



1. FTP port mode sends a PORT message:

CLAT: The source address/port is translated from lpv4/port1 to lpv6/port1.

PLAT: The source address/port is lpv6_1/port1 is translated to lpv4_1'/port1' and matches the NAT64 rule. The lpv4_1/port2 in the payload is translated to lpv4_1'/port2' and the FTP ALG creates a gate such as lpv4_1'/port2'(lpv4_1/port2->server_ip/server_port).

2. The first packet reaches the gate from the server side: When the traffic reaches the gate, the first packet from the server side with destination lpv4_1'/port2' is translated to lpv6_1/port2. Finally, the packet reaches the client lpv4_1/port2 before CLAT.

Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
12.3X48-D15	

RELATED DOCUMENTATION

Understanding IPv6 ALG Support for ICMP | 176

Understanding ALG Support for VRF Routing Instance

To support MPLS-based implementations, virtual routing and forwarding (VRF) instances are supported in an Application Layer Gateway (ALG) module. The ALG module is responsible for associating multiple connections from an application with the initial session that the application creates. The ALG module intercepts and analyzes the specified traffic, allocates resources, and defines dynamic policies to permit the traffic to pass securely through the device.

An ALG module performs the following functions:

- Inspects the packet for an embedded IP address and port information in the packet payload. When the first packet arrives at the device, it undergoes flow first path processing to identify whether the incoming traffic could match gate, the search key uses details such as zone, source IP address, destination IP address, source port, destination port, source VRF details and destination VRF details.
- Opens a pinhole for a new connection between a client and a server, and transfers data between a client and a server located on opposite sides.
- Performs Network Address Translation (NAT) processing, if necessary.

The ALG module also opens a gate for the IP address and port number to permit data exchange for the control and data sessions. Starting in Junos OS Release 15.1X49-D160, ALG supports control sessions and data sessions belonging to the same VRF.

RELATED DOCUMENTATION

Configuring Security Policies for a VRF Routing Instance

NAT for VRF Routing Instance

[Flow Management in SRX Series Devices Using VRF Routing Instance](#)

3

CHAPTER

VoIP ALGs

IN THIS CHAPTER

- Understanding VoIP ALG Types | 188
 - VoIP DSCP Rewrite Rules | 189
 - H.323 ALG | 192
 - MGCP ALG | 247
 - SCCP ALG | 294
 - SIP ALG | 326
-

Understanding VoIP ALG Types

Junos OS supports voice-over-IP Application Layer Gateways (VoIP ALGs) and basic data ALGs. (Note that supported ALG types vary depending on which hardware device you are using.)

VoIP ALGs provide stateful Application Layer inspection and Network Address Translation (NAT) capabilities to VoIP signaling and media traffic. The ALG inspects the state of transactions, or calls, and forwards or drops packets based on those states.

Junos OS supports the following VoIP ALGs:

[Table 5 on page 188](#) lists the VoIP ALG types.

Table 5: VoIP-ALG-Types

VoIP ALG	Description
H.323	The H.323 ALG provides support for the H.323 legacy VoIP protocol. The ALG lets you secure VoIP communication between terminal hosts, such as IP phones and multimedia devices. In such a telephony system, the gatekeeper device manages call registration, admission, and call status for VoIP calls. Gatekeepers can reside in the two different zones or in the same zone.
MGCP	The MGCP ALG provides support for Media Gateway Control Protocol (MGCP). MGCP is a text-based Application Layer protocol used for call setup and call control between the media gateway and the media gateway controller (MGC).
SCCP	The SCCP ALG provides support for Skinny Client Control Protocol (SCCP). SCCP is a Cisco proprietary protocol for call signaling. Skinny is based on a call-agent-based call-control architecture. The control protocol uses binary-coded frames encoded on TCP frames sent to well-known TCP port number destinations to set up and tear down RTP media sessions.
SIP	The SIP ALG provides support for the Session Initiation Protocol (SIP). SIP is an Internet Engineering Task Force (IETF)-standard protocol for initiating, modifying, and terminating multimedia sessions over the Internet. Such sessions might include conferencing, telephony, or multimedia, with features such as instant messaging and application-level mobility in network environments.

For information about enabling and configuring each of these ALGs through J-Web, select the **Configure>Security>ALG** page in the J-Web user interface and click **Help**.

RELATED DOCUMENTATION

- [ALG Overview | 2](#)
- [Understanding the IKE and ESP ALG | 38](#)
- [Understanding H.323 ALG | 192](#)
- [Understanding the SIP ALG | 327](#)
- [Understanding SCCP ALGs | 295](#)
- [Understanding the MGCP ALG | 247](#)
- [Understanding RPC ALGs | 72](#)

VoIP DSCP Rewrite Rules

IN THIS SECTION

- [Understanding VoIP DSCP Rewrite Rules | 189](#)
- [Example: Configuring VoIP DSCP Rewrite Rules | 190](#)

The Vo IP rewrite rules modifies the appropriate *class of service* (CoS) bits in an outgoing packets through Differentiated Services Code Point (DSCP) mechanism that improves the VoIP quality in a congested network.

Understanding VoIP DSCP Rewrite Rules

This topic describes the voice over IP Application Layer Gateway (VoIP ALG) mechanism for modifying the Differentiated Services Code Point (DSCP) field of Real-Time Transport Protocol (RTP) packets. The VoIP ALG mechanism is applicable for the RTP session, which is recognized by the ALG.

DSCP is a modification of the type of service byte for *class of service* (CoS). Six bits of this byte are reallocated for use as the DSCP field, where each DSCP specifies a particular per-hop behavior that is applied to a packet.

A rewrite rule modifies the appropriate CoS bits in an outgoing packet to meet the requirements of the targeted peer. Each rewrite rule reads the current CoS value that is configured at the VoIP ALG level. Every packet that hits the VoIP ALG is marked by this CoS value.

This feature supports ALG DSCP marking for H323, Session Initiation Protocol (SIP), Media Gateway Control Protocol (MGCP), and Skinny Client Control Protocol (SCCP). It provides a 6-bit DSCP value configuration for each of these. When the first RTP packet hits the ALG, this feature receives the 6-bit DSCP value from the configuration and sets it to the RTP session that the packet has created. This first RTP packet and the following RTP packets passing through the RTP session are marked according to the 6-bit DSCP value in the session.

To avoid VoIP quality degradation caused by network congestion, the RTP packets are required to mark the DSCP bit to ensure they get higher routing priority. A downstream router can put those packets in a higher priority queue for faster forwarding. To provide this functionality, there needs to be a per-VoIP mechanism for modifying the DSCP field of RTP packets according to the specific configuration. This will ensure that all RTP packets based on User Datagram Protocol/Transport Control Protocol (UDP/TCP) that encounter the ALG will be assigned a specific DSCP bit.

Example: Configuring VoIP DSCP Rewrite Rules

IN THIS SECTION

- [Requirements | 190](#)
- [Overview | 190](#)
- [Configuration | 191](#)
- [Verification | 191](#)

This example shows how to configure VoIP DSCP.

Requirements

This example assumes that the VoIP ALG has been enabled. (Platform support depends on the Junos OS release in your installation.)

Overview

This example shows how to configure four ALG DSCP markings; SIP, H323, MGCP, and SCCP. You set the 6-bit DSCP value configuration for each ALG DSCP.

Configuration

IN THIS SECTION

- [Procedure | 191](#)

Procedure

Step-by-Step Procedure

To configure VoIP DSCP rewrite rules:

1. Set the DSCP for each VoIP ALG.

```
[edit]
user@host# set security alg sip dscp-rewrite code-point 101010
user@host# set security alg h323 dscp-rewrite code-point 010101
user@host# set security alg mgcp dscp-rewrite code-point 111000
user@host# set security alg sccp dscp-rewrite code-point 000111
```

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

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

Verification

To verify that the configuration is working properly, enter the `show security alg` command.

RELATED DOCUMENTATION

[MGCP ALG | 247](#)

[SCCP ALG | 294](#)

[SIP ALG | 326](#)

H.323 ALG

IN THIS SECTION

- [Understanding H.323 ALG | 192](#)
- [Understanding the Avaya H.323 ALG | 195](#)
- [Example: Passing H.323 ALG Traffic to a Gatekeeper in the Private Zone | 197](#)
- [Example: Passing H.323 ALG Traffic to a Gatekeeper in the External Zone | 204](#)
- [Example: Using NAT with the H.323 ALG to Enable Incoming Calls | 211](#)
- [Example: Using NAT with the H.323 ALG to Enable Outgoing Calls | 222](#)
- [Example: Setting H.323 ALG Endpoint Registration Timeouts | 232](#)
- [Example: Setting H.323 ALG Media Source Port Ranges | 234](#)
- [Example: Configuring H.323 ALG DoS Attack Protection | 236](#)
- [Understanding H.323 ALG Known Message Types | 238](#)
- [Understanding H.323 ALG Unknown Message Types | 243](#)
- [Example: Allowing Unknown H.323 ALG Message Types | 244](#)

The H.323 Application Layer Gateway (ALG) consist of a suite of H.225.0 and H.245 protocols to provide audio-visual communication session on any network. The H.323 ALG provides a secure communication between terminal hosts, such as IP phones and multimedia devices.

Understanding H.323 ALG

IN THIS SECTION

- [H.323 ALG Configuration Overview | 195](#)

The H.323 standard is a legacy voice-over-IP (VoIP) protocol defined by the International Telecommunication Union (ITU-T). H.323 consists of a suite of protocols (such as H.225.0 and H.245) that are used for call signaling and call control for VoIP.

H.323 uses the ASN.1 coding format. It sets up the dynamic links for data, video, and audio streams, following the protocols Q.931 (with port number 1720) and H.245. There are three major processes in H.323:

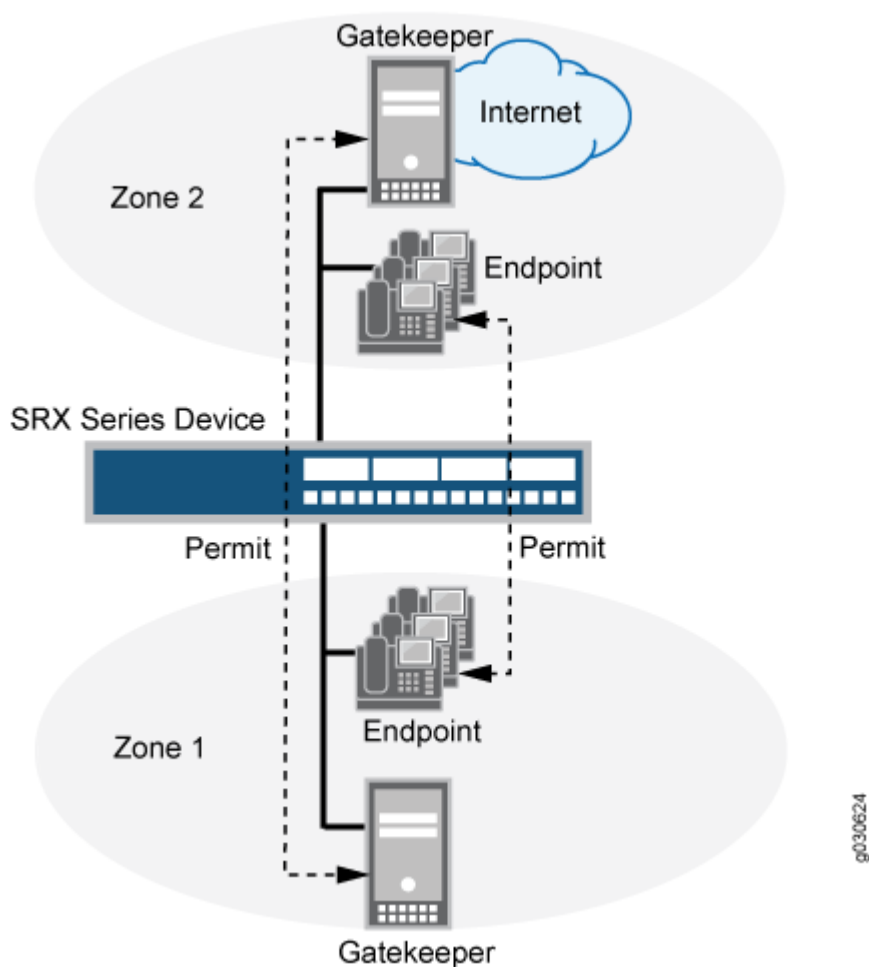
- **Gatekeeper Discovery**—An endpoint finds its gatekeeper through the gatekeeper discovery process, via broadcast or unicast to a known IP and the well-known UDP port 1719. Note that Junos OS only supports unicast discovery.
- **Endpoint Registration, Admission, and Status**—An endpoint registers to a gatekeeper and asks for its management. Before making a call, an endpoint asks its gatekeeper for permission to place the call. In both registration and admission phases, the Registration, Admission, and Status (RAS) channel is used. The Transport Service Access Point (TSAP) can utilize either the well-known UDP port (1719) or a dynamically assigned port from the discovery or registration phase and an IP address.
- **Call Control and Call Setup**—Calls can be established within a zone or across two zones, or even across multiple zones (multipoint conference). The call setup and tear down is performed through the call signaling channel whose TSAP is the well-known TCP port (1720). The call control, including opening/closing media channels between two endpoints, is performed through the call control channel whose TSAP is dynamically assigned from the previous call signaling process. H.245 messages are used in the call control channel, and are encoded using ASN.1.



NOTE: Detailed information on H.323 can be found in ITU-T Recommendation H.323.

The H.323 Application Layer Gateway (ALG) of the device lets you secure VoIP communication between terminal hosts, such as IP phones and multimedia devices. In such a telephony system, the gatekeeper device manages call registration, admission, and call status for VoIP calls. Gatekeepers can reside in the two different zones or in the same zone. (See [Figure 18 on page 194](#).)

Figure 18: H.323 ALG for VoIP Calls



NOTE: The illustration uses IP phones for illustrative purposes, although it is possible to make configurations for other hosts that use VoIP, such as Microsoft NetMeeting multimedia devices.

Starting with Junos OS Release 17.4R1, the gateway-to-gateway call feature is supported on the H.323 Application Layer Gateway (ALG). This feature introduces one-to-many mapping between an H.225 control session and H.323 calls as multiple H.323 calls go through a single control session.

Starting with Junos OS Release 17.4R1, the H.323 Application Layer Gateway (ALG) supports NAT64 rules in an IPv6 network.

H.323 ALG Configuration Overview

The H.323 Application Layer Gateway (ALG) is enabled by default on the device—no action is required to enable it. However, you might choose to fine-tune H.323 ALG operations by using the following instructions:

1. Specify how long an endpoint registration entry remains in the Network Address Translation (NAT) table. For instructions, see ["Example: Setting H.323 ALG Endpoint Registration Timeouts" on page 232](#).
2. Enable media traffic on a narrow or wide range of ports. For instructions, see ["Example: Setting H.323 ALG Media Source Port Ranges" on page 234](#).
3. Protect the H.323 gatekeeper from denial-of-service (DoS) flood attacks. For instructions, see ["Example: Configuring H.323 ALG DoS Attack Protection" on page 236](#).
4. Enable unknown messages to pass when the session is in NAT mode and route mode. For instructions, see ["Example: Allowing Unknown H.323 ALG Message Types" on page 244](#).

Understanding the Avaya H.323 ALG

IN THIS SECTION

- [Avaya H.323 ALG-Specific Features | 196](#)
- [Call Flow Details in the Avaya H.323 ALG | 196](#)

The H.323 standard is a legacy voice-over-IP (VoIP) protocol defined by the International Telecommunication Union (ITU-T). H.323 consists of a suite of protocols (such as H.225.0 and H.245) that are used for call signaling and call control for VoIP. The processes for configuring the H.323 standard Application Layer Gateway (ALG) and the proprietary Avaya H.323 ALG are the same.

However, Avaya H.323 ALG has some special features. To understand and configure the Avaya H.323-specific features listed here, see the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

This topic contains the following sections:

Avaya H.323 ALG-Specific Features

Avaya H.323-specific features are as follows:

- H.323 Fast Connect
- H.323 asymmetric media
- Call waiting
- Call forwarding
- Voice mail
- Call identification
- Conference calling

Call Flow Details in the Avaya H.323 ALG

- Connecting the Phone into the Network—Avaya performs the Q.931 Setup/Connect negotiation when the phone is wired into the network rather than when a call is being initiated.
- Making a call—When a call is made, because the PBX has already stored the capabilities for each phone when the phone is connected to the network, no further Q.931 and PBX negotiations are required to set up the call. It no longer exchanges Q.931 Setup and Connect messages with the PBX. The phone and the PBX exchange H.323 Facility messages to set up the call.
- Registering with a CM—When a call has been made, Avaya H.323 registers with the Avaya Communication Manager (CM). The registration process is similar to a generic H.323 standard registration process.



NOTE: The direct mode and tunnel mode are not defined by Avaya H.323 ALG.

For a call to work, the CM must be deployed with Avaya Endpoints. During the call, RAS and Q.931 messages are exchanged between the CM and the Avaya Endpoints.



NOTE: For Avaya H.323 with a source Network Address Translation (NAT) pool, the registration process allows only one IP address in the pool.

- Setting up Real-Time Transport Protocol (RTP)/Real-Time Control Protocol (RTCP) ports—The Q.931 Setup, Facility and Information messages are used to set up RTP/RTCP ports. The hierarchy for an Avaya H.323 session is Q.931, RTP/RTCP, Parent, and then Child.



NOTE: H.245 ports are not used in an Avaya call flow process.

- Using Avaya H.323 counters—The counters for calls and active calls are not applicable to the Avaya H.323 ALG. The call creation and tearing down is done by Facility messages afterward. When resources are allocated for a call, all counters for calls and active calls increment. If resources are allocated for a call multiple times, messages belonging to the same call that pass the firewall multiple times will trigger multiple increments of the counters. In other words, messages that belong to the same call and pass the firewall multiple times might trigger multiple increments of the counters if the resource for a call needs to be allocated multiple times.

For example, in the two-zone case, the setup and connect message pair allocates one call resource. The active call counter is increased once. Each time the setup and connect message pair passes the firewall, a different call resource with unique interfaces and NAT is allocated. Therefore, the counter increments twice in a three-zone scenario.

Example: Passing H.323 ALG Traffic to a Gatekeeper in the Private Zone

IN THIS SECTION

- Requirements | 197
- Overview | 198
- Configuration | 198
- Verification | 202

This example shows how to set up two policies that allow H.323 traffic to pass between IP phone hosts and a gatekeeper in the private zone, and an IP phone host (2.2.2.5/32) in the public zone.

Requirements

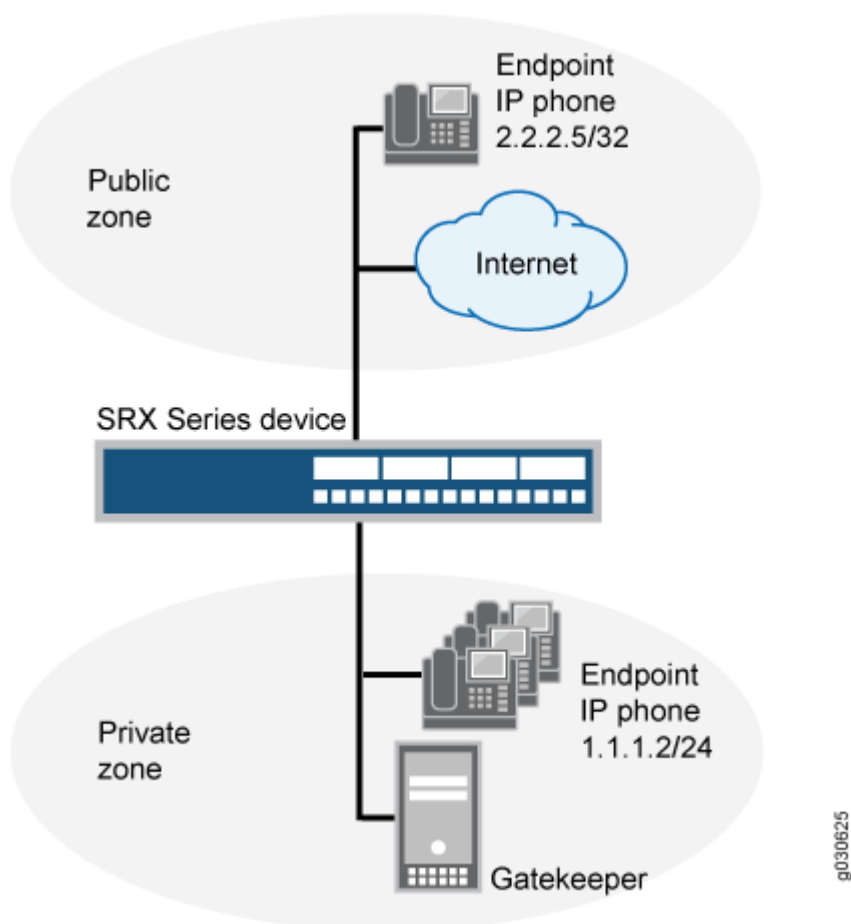
Before you begin:

- Understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.
- Configure security zones. See *Understanding Security Zones*.

Overview

This example shows how to set up two policies that allow H.323 traffic to pass between IP phone hosts and a gatekeeper in the private zone, and an IP phone host (2.2.2.5/32) in the public zone. The connection to the device can either be with or without NAT. See [Figure 19 on page 198](#).

Figure 19: H.323 Gatekeeper in the Private Zone



Configuration

IN THIS SECTION

- [Procedure | 199](#)

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security zones security-zone public address-book address ip_phone 2.2.2.5/32
set security zones security-zone private address-book address gateway 2.2.2.5/32
set security policies from-zone private to-zone public policy P1 match source-address any
set security policies from-zone private to-zone public policy P1 match destination-address
IP_Phone
set security policies from-zone private to-zone public policy P1 match application junos-h323
set security policies from-zone private to-zone public policy P1 then permit
set security policies from-zone public to-zone private policy P2 match source-address any
set security policies from-zone public to-zone private policy P2 match destination-address
gateway
set security policies from-zone public to-zone private policy P2 match application junos-h323
set security policies from-zone public to-zone private policy P2 then permit
```

Step-by-Step Procedure

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

To configure the device to pass H.323 ALG traffic to a gatekeeper in the private zone:

1. Configure two address books.

```
[edit]
user@host# set security zones security-zone public address-book address ip_phone 2.2.2.5/32
set security zones security-zone private address-book address gateway 2.2.2.5/32
```

2. Configure policy P1 from the private zone to the public zone.

```
[edit]
user@host# set security policies from-zone private to-zone public policy P1 match source-
address any
```

```

user@host# set security policies from-zone private to-zone public policy P1 match destination-
address IP_Phone
user@host# set security policies from-zone private to-zone public policy P1 match application
junos-h323
user@host# set security policies from-zone private to-zone public policy P1 then permit

```

3. Configure policy P2 from the public zone to the private zone.

```

[edit]
user@host# set security policies from-zone public to-zone private policy P2 match source-
address any
user@host# set security policies from-zone public to-zone private policy P2 match destination-
address gateway
user@host# set security policies from-zone public to-zone private policy P2 match application
junos-h323
user@host# set security policies from-zone public to-zone private policy P2 then permit

```

Results

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

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

```

[edit]
user@host# show security policies
...
from-zone trust to-zone trust {
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
}

```

```

from-zone trust to-zone untrust {
  policy default-permit {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}

from-zone untrust to-zone trust {
  policy default-deny {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      deny;
    }
  }
}

from-zone private to-zone public {
  policy P1 {
    match {
      source-address any;
      destination-address IP_Phone;
      application junos-h323;
    }
    then {
      permit;
    }
  }
}

from-zone public to-zone private {
  policy P2 {
    match {
      source-address any;
      destination-address gateway;
      application junos-h323;
    }
  }
}

```

```

        then {
            permit;
        }
    }
}
...

```

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

Verification

IN THIS SECTION

- [Verifying H.323 ALG Configurations | 202](#)

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

Verifying H.323 ALG Configurations

Purpose

Display information about active calls.



NOTE: H.323 counters for calls and active calls in the output to this `show security` command do not apply to the proprietary Avaya implementation of H.323. This is because Q.931 setup and connect messages are exchanged right after the phone is powered up and call creation and tear down is done by Facility messages.

Counters for calls and active calls are increased when the resources allocated for calls are increased—that is, messages belonging to the same call and that pass the firewall multiple times increment the counters. This applies when resources for a call need to be allocated multiple times. For example, in a two-zone scenario the setup and connect message pair allocates one call resource, and the active call counter is increased by one. But in a three-zone scenario the setup and connect message pair passes the firewall twice, each time allocating different call resources. In this case, the counter is incremented.

Action

From the J-Web interface, select Monitor>ALGs>H323. Alternatively, from the CLI, enter the `show security alg h323 counters` command.

Counters for H.245 messages received also will not be accurate in the case of H.245 tunneling. Because H.245 messages are encapsulated in Q.931 packets, the counter for H.245 messages received will remain zero even when there are H.245 messages. The other H245 counter will, however, reflect these packet transmissions.

```
[edit]
user@host> show security alg h323 counters
H.323 counters summary:
  Packets received      : 0
  Packets dropped       : 0
  RAS message received  : 0
  Q.931 message received : 0
  H.245 message received : 0
  Number of calls       : 0
  Number of active calls : 0
H.323 error counters:
  Decoding errors       : 0
  Message flood dropped  : 0
  NAT errors            : 0
  Resource manager errors : 0
H.323 message counters:
  RRQ      : 0
  RCF      : 0
  ARQ      : 0
  ACF      : 0
  URQ      : 0
  UCF      : 0
  DRQ      : 0
  DCF      : 0
  Oth RAS  : 0
  Setup    : 0
  Alert    : 0
  Connect  : 0
  CallProd : 0
  Info     : 0
  RelCmpl  : 0
  Facility : 0
```



```

Empty      : 0
OLC        : 0
OLC-ACK    : 0
Oth H245   : 0

```

Example: Passing H.323 ALG Traffic to a Gatekeeper in the External Zone

IN THIS SECTION

- Requirements | 204
- Overview | 204
- Configuration | 205
- Verification | 209

This example shows how to set up two policies to allow H.323 traffic to pass between IP phone hosts in the internal zone, and the IP phone at IP address 2.2.2.5/32 (and the gatekeeper) in the external zone.

Requirements

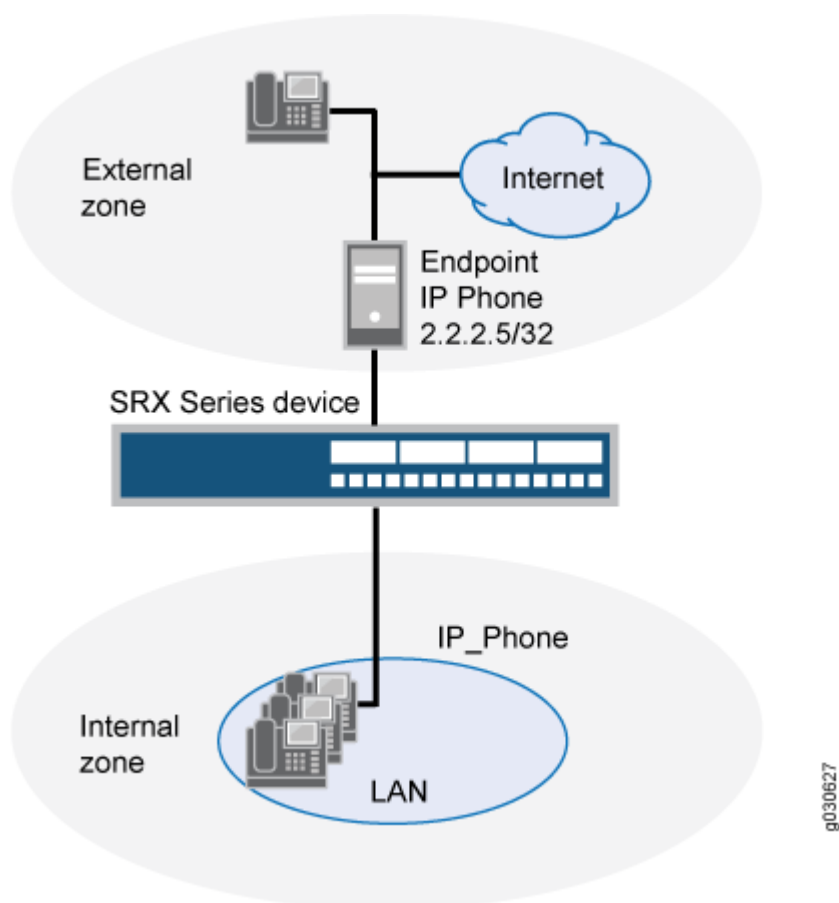
Before you begin:

- Understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.
- Configure security zones. See *Understanding Security Zones*.

Overview

Because route mode does not require address mapping of any kind, a device configuration for a gatekeeper in the external, or public, zone is usually identical to the configuration for a gatekeeper in an internal, or private, zone. This example shows how to set up two policies to allow H.323 traffic to pass between IP phone hosts in the internal zone, and the IP phone at IP address 2.2.2.5/32 (and the gatekeeper) in the external zone. The device can be in transparent or route mode. See [Figure 20 on page 205](#).

Figure 20: H.323 Gatekeeper in the External Zone



Configuration

IN THIS SECTION

- [Procedure | 205](#)

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy

and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security zones security-zone external address-book address IP_Phone 2.2.2.5/32
set security zones security-zone internal address-book address gatekeeper 2.2.2.10/32
set security policies from-zone internal to-zone external policy P1 match source-address any
set security policies from-zone internal to-zone external policy P1 match destination-address
IP_Phone
set security policies from-zone internal to-zone external policy P1 match application junos-h323
set security policies from-zone internal to-zone external policy P1 then permit
set security policies from-zone internal to-zone external policy P2 match source-address any
set security policies from-zone internal to-zone external policy P2 match destination-address
gatekeeper
set security policies from-zone internal to-zone external policy P2 match application junos-h323
set security policies from-zone internal to-zone external policy P2 then permit
set security policies from-zone external to-zone internal policy P3 match source-address IP_Phone
set security policies from-zone external to-zone internal policy P3 match destination-address any
set security policies from-zone external to-zone internal policy P3 match application junos-h323
set security policies from-zone external to-zone internal policy P3 then permit
set security policies from-zone external to-zone internal policy P4 match source-address
gatekeeper
set security policies from-zone external to-zone internal policy P4 match destination-address any
set security policies from-zone external to-zone internal policy P4 match application junos-h323
set security policies from-zone external to-zone internal policy P4 then permit
```

Step-by-Step Procedure

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

To configure the device to pass H.323 ALG traffic to a gatekeeper in the external zone:

1. Configure two address books.

```
[edit]
user@host# set security zones security-zone external address-book address IP_Phone 2.2.2.5/32
user@host# set security zones security-zone internal address-book address gatekeeper
2.2.2.10/32
```

2. Configure policy P1 from the internal zone to the external zone.

```
[edit]
user@host# set security policies from-zone internal to-zone external policy P1 match source-
address any
user@host# set security policies from-zone internal to-zone external policy P1 match
destination-address IP_Phone
user@host# set security policies from-zone internal to-zone external policy P1 match
application junos-h323
user@host# set security policies from-zone internal to-zone external policy P1 then permit
```

3. Configure policy P2 to allow traffic between the internal zone and the gatekeeper in the external zone.

```
[edit]
user@host# set security policies from-zone internal to-zone external policy P2 match source-
address any
user@host# set security policies from-zone internal to-zone external policy P2 match
destination-address gatekeeper
user@host# set security policies from-zone internal to-zone external policy P2 match
application junos-h323
user@host# set security policies from-zone internal to-zone external policy P2 then permit
```

4. Configure policy P3 to allow traffic between phones in the internal zone and the external zone.

```
[edit]
user@host# set security policies from-zone external to-zone internal policy P3 match source-
address IP_Phone
user@host# set security policies from-zone external to-zone internal policy P3 match
destination-address any
user@host# set security policies from-zone external to-zone internal policy P3 match
application junos-h323
user@host# set security policies from-zone external to-zone internal policy P3 then permit
```

5. Configure policy P4 to allow traffic between phones in the internal zone and the gatekeeper in the external zone.

```
[edit]
user@host# set security policies from-zone external to-zone internal policy P4 match source-
```

```

address gatekeeper
user@host# set security policies from-zone external to-zone internal policy P4 match
destination-address any
user@host# set security policies from-zone external to-zone internal policy P4 match
application junos-h323
user@host# set security policies from-zone external to-zone internal policy P4 then permit

```

Results

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

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

```

[edit]

user@host# show security policies
...
from-zone internal to-zone external {
  policy P1 {
    match {
      source-address any;
      destination-address IP_Phone;
      application junos-h323;
    }
    then {
      permit;
    }
  }
  policy P2 {
    match {
      source-address any;
      destination-address gatekeeper;
      application junos-h323;
    }
    then {
      permit;
    }
  }
}
}

```

```

from-zone external to-zone internal {
  policy P3 {
    match {
      source-address IP_Phone;
      destination-address any;
      application junos-h323;
    }
    then {
      permit;
    }
  }
  policy P4 {
    match {
      source-address gatekeeper;
      destination-address any;
      application junos-h323;
    }
    then {
      permit;
    }
  }
}
...

```

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

Verification

IN THIS SECTION

- [Verifying H.323 ALG Configurations | 209](#)

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

Verifying H.323 ALG Configurations

Purpose

Display information about active calls.



NOTE: H.323 counters for calls and active calls in the output to this `show security` command do not apply to the proprietary Avaya implementation of H.323. This is because Q.931 setup and connect messages are exchanged right after the phone is powered up and call creation and tear down is done by Facility messages.

Counters for calls and active calls are increased when the resources allocated for calls are increased—that is, messages belonging to the same call and that pass the firewall multiple times increment the counters. This applies when resources for a call need to be allocated multiple times. For example, in a two-zone scenario the setup and connect message pair allocates one call resource, and the active call counter is increased by one. But in a three-zone scenario the setup and connect message pair passes the firewall twice, each time allocating different call resources. In this case, the counter is incremented.

Action

From the J-Web interface, select `Monitor>ALGs>H323`. Alternatively, from the CLI, enter the `show security alg h323 counters` command.

Counters for H.245 messages received also will not be accurate in the case of H.245 tunneling. Because H.245 messages are encapsulated in Q.931 packets, the counter for H.245 messages received will remain zero even when there are H.245 messages. The `Other H245` counter will, however, reflect these packet transmissions.

```
[edit]
user@host> show security alg h323 counters
H.323 counters summary:
  Packets received      : 0
  Packets dropped       : 0
  RAS message received  : 0
  Q.931 message received : 0
  H.245 message received : 0
  Number of calls       : 0
  Number of active calls : 0
H.323 error counters:
  Decoding errors       : 0
  Message flood dropped  : 0
  NAT errors            : 0
  Resource manager errors : 0
H.323 message counters:
  RRQ                   : 0
```

```
RCF      : 0
ARQ      : 0
ACF      : 0
URQ      : 0
UCF      : 0
DRQ      : 0
DCF      : 0
Oth RAS  : 0
Setup    : 0
Alert    : 0
Connect  : 0
CallProd : 0
Info     : 0
RelCmpl  : 0
Facility : 0
Empty    : 0
OLC      : 0
OLC-ACK  : 0
Oth H245 : 0
```

Example: Using NAT with the H.323 ALG to Enable Incoming Calls

IN THIS SECTION

- [Requirements | 211](#)
- [Overview | 212](#)
- [Configuration | 213](#)
- [Verification | 219](#)

This example shows how to configure NAT with the H.323 ALG to enable calls from a public to a private network.

Requirements

Before you begin, understand H.323 ALGs. See ["Understanding H.323 ALG" on page 192](#).

Overview

IN THIS SECTION

- [Topology | 212](#)

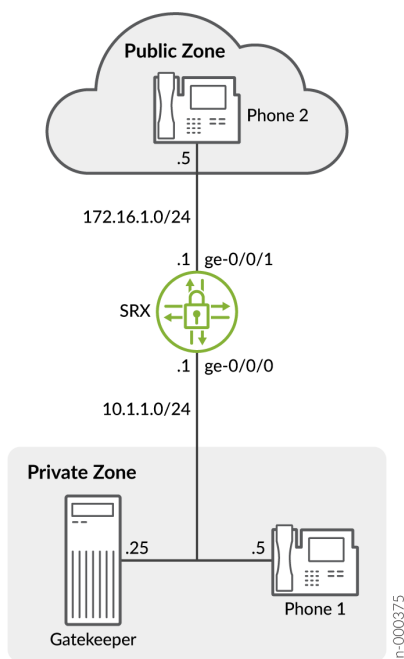
In a two-zone scenario with a server in the private zone, you can use NAT for incoming calls by configuring a NAT pool on the interface to the public zone.

In this example (see [Figure 21 on page 212](#)), IP-Phone1 and a server called gatekeeper are in the private zone, and IP-Phone2 is in the public zone. You configure a static nat rule set and a source NAT pool to do NAT. You also create two policies, private-to-public and public-to-private, to permit ALG H.323 traffic from and to the private and public zones.

Topology

[Figure 21 on page 212](#) shows NAT with the H.323 ALG incoming calls.

Figure 21: NAT with the H.323 ALG—Incoming Calls



In this example, you configure source NAT as follows:

- Create a static NAT rule set called `gatekeeper` with a rule called `gatekeeper` to match packets from the public zone with the destination address `172.16.1.25/32`. For matching packets, the destination IP address is translated to the private address `10.1.1.25/32`.
- Define a source NAT pool called `h323-nat-pool` to contain the IP address range from `172.16.1.30/32` through `172.16.1.150/32`.
- Create a source NAT rule set called `h323-nat` with rule `h323-r1` to match packets from the private zone to the public zone with the source IP address `10.1.1.0/24`. For matching packets, the source address is translated to the IP address in `h323-nat-pool`.
- Configure proxy ARP for the addresses `172.16.1.30/32` through `172.16.1.150/32` on interface `ge-0/0/1.0`. This allows the system to respond to ARP requests received on the interface for these addresses.

Configuration

IN THIS SECTION

- [Procedure | 213](#)

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the `[edit]` hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24
set security zones security-zone private address-book address IP-Phone1 10.1.1.5/32
set security zones security-zone private address-book address gatekeeper 10.1.1.25/32
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address IP-Phone2 172.16.1.5/32
set security zones security-zone public interfaces ge-0/0/1.0
```

```

set security nat source pool h323-nat-pool address 172.16.1.30/32 to 172.16.1.150/32
set security nat source address-persistent
set security nat source rule-set h323-nat from zone private
set security nat source rule-set h323-nat to zone public
set security nat source rule-set h323-nat rule h323-r1 match source-address 10.1.1.0/24
set security nat source rule-set h323-nat rule h323-r1 then source-nat pool h323-nat-pool
set security nat proxy-arp interface ge-0/0/1.0 address 172.16.1.30/32 to 172.16.1.150/32
set security policies from-zone private to-zone public policy private-to-public match source-
address IP-Phone1
set security policies from-zone private to-zone public policy private-to-public match source-
address gatekeeper
set security policies from-zone private to-zone public policy private-to-public match
destination-address IP-Phone2
set security policies from-zone private to-zone public policy private-to-public match
application junos-h323
set security policies from-zone private to-zone public policy private-to-public then permit
set security policies from-zone public to-zone private policy public-to-private match source-
address IP-Phone2
set security policies from-zone public to-zone private policy public-to-private match
destination-address IP-Phone1
set security policies from-zone public to-zone private policy public-to-private match
destination-address gatekeeper
set security policies from-zone public to-zone private policy public-to-private match
application junos-h323
set security policies from-zone public to-zone private policy public-to-private then permit

```

Step-by-Step Procedure

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

To configure NAT with H.323 ALG to enable calls from a public to a private network:

1. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24

```

2. Configure zones and assign addresses to them.

```
[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone private address-book address IP-Phone1 10.1.1.5/32
user@host# set security-zone private address-book address gatekeeper 10.1.1.25/32
user@host# set security-zone public interfaces ge-0/0/1.0
user@host# set security-zone public address-book address IP-Phone2 172.16.1.5/32
```

3. Create a static NAT rule set.

```
[edit security nat static rule-set ip-phones]
user@host# set from zone public
user@host# set match destination-address 172.16.1.25/32
user@host# set then static-nat prefix 10.1.1.25/32
```

4. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/1.0 address 172.16.1.25/32
```

5. Configure a source NAT rule set.

```
[edit security nat]
set source pool h323-nat-pool address 172.16.1.30/32 to 172.16.1.150/32
set source address-persistent
set source rule-set h323-nat from zone private
set source rule-set h323-nat to zone public
set source rule-set h323-nat rule h323-r1 match source-address 10.1.1.0/24
set source rule-set h323-nat rule h323-r1 then source-nat pool h323-nat-pool
set proxy-arp interface ge-0/0/1.0 address 171.16.1.30/32 to 172.16.1.150/32
```

6. Configure policies for outgoing traffic.

```
[edit security policies from-zone private to-zone public policy private-to-public]
user@host# set match source-address IP-Phone1
user@host# set match source-address gatekeeper
user@host# set match destination-address IP-Phone2
```

```

user@host# set match application junos-h323
user@host# set then permit

```

7. Configure policies for incoming traffic.

```

[edit security policies from-zone public to-zone private policy public-to-private]
user@host# set match source-address IP-Phone2
user@host# set match destination-address IP-Phone1
user@host# set match destination-address gatekeeper
user@host# set match application junos-h323
user@host# set then permit

```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, `show security nat`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show interfaces
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.1.1.1/24;
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family inet {
                address 172.16.1.1/24;
            }
        }
    }
[edit]
user@host# show security zones
security-zone private {
    address-book {
        address IP-Phone1 10.1.1.5/32;
        address gatekeeper 10.1.1.25/32;
    }
}

```

```

    }
    interfaces {
        ge-0/0/0.0;
    }
}

security-zone public {
    address-book {
        address IP-Phone2 172.16.1.5/32;
    }
    interfaces {
        ge-0/0/1.0;
    }
}

[edit]
user@host# show security nat
source {
    pool h323-nat-pool {
        address {
            172.16.1.30/32 to 172.16.1.150/32;
        }
    }
    address-persistent;
    rule-set h323-nat {
        from zone private;
        to zone public;
        rule h323-r1 {
            match {
                source-address 10.1.1.0/24;
            }
            then {
                source-nat {
                    pool {
                        h323-nat-pool;
                    }
                }
            }
        }
    }
}

proxy-arp {
    interface ge-0/0/1.0 {
        address {
            172.16.1.30/32 to 172.16.1.150/32;

```

```

    }
  }
}
static {
  rule-set ip-phones {
    from zone public;
    rule gatekeeper {
      match {
        destination-address 172.16.1.25/32;
      }
      then {
        static-nat prefix 10.1.1.25/32;
      }
    }
  }
}
}
proxy-arp {
  interface ge-0/0/1.0 {

    address {
      172.16.1.25/32;
    }
  }
}
}
[edit]
user@host# show security policies
from-zone private to-zone public {
  policy private-to-public {
    match {
      source-address [IP-Phone1 gatekeeper];
      destination-address IP-Phone2;
      application junos-h323;
    }
    then {
      permit;
    }
  }
}
}
from-zone public to-zone private {
  policy public-to-private {
    match {
      source-address IP-Phone2;
      destination-address [IP-Phone1 gatekeeper];
    }
  }
}
}

```

```
        application junos-h323;  
    }  
    then {  
        permit;  
    }  
}  
}
```

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

Verification

IN THIS SECTION

- [Verifying H.323 ALG Status | 219](#)
- [Verifying Security ALG H.323 Counters | 220](#)
- [Verifying Source NAT Rule Usage | 221](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying H.323 ALG Status

Purpose

Verify that H.323 ALG is enabled on your system.

Action

From operational mode, enter the `show security alg status` command.

```
user@host> show security alg status  
ALG Status :  
  DNS      : Enabled  
  FTP      : Enabled  
  H323     : Enabled  
  MGCP     : Enabled  
  MSRPC    : Enabled  
  PPTP     : Enabled
```



```

RSH      : Disabled
RTSP     : Enabled
SCCP     : Enabled
SIP      : Enabled
SQL      : Enabled
SUNRPC   : Enabled
TALK     : Enabled
TFTP     : Enabled
IKE-ESP  : Disabled

```

Meaning

The output shows the H323 ALG status as follows:

- Enabled—Shows the H323 ALG is enabled.
- Disabled—Shows the H323 ALG is disabled.

Verifying Security ALG H.323 Counters

Purpose

Verify that there is a security counters for ALG H323.

Action

From operational mode, enter the `show security alg h323 counters` command.

```
user@host> show security alg h323 counters
```

```

H.323 counters summary:
Packets received :4060
Packets dropped  :24
RAS message received :3690
Q.931 message received :202
H.245 message received :145
Number of calls :25
Number of active calls :0

H.323 Error Counters:

```

```

Decoding errors :24
Message flood dropped :0
NAT errors :0
Resource manager errors :0

```

H.323 Message Counters:

```

RRQ : 431
RCF : 49
ARQ : 60
ACF : 33
URQ : 34
UCF : 25
DRQ : 55
DCF : 44
oth RAS : 2942
Setup : 28
Alert : 9
Connect : 25
CallPrcd : 18
Info : 0
RelCmpl : 39
Facility : 14
Progress : 0
Empty : 65
OLC : 20
OLC-ACK : 20

```

Meaning

The sample output gives the rundown of security ALG H.323 counters expressing that, there are security counters for ALG H323.

Verifying Source NAT Rule Usage

Purpose

Verify that there is traffic matching the source NAT rule.

Action

From operational mode, enter the `show security nat source rule all` command. View the Translation hits field to check for traffic that matches the rule.

```
user@host> show security nat source rule all
source NAT rule: h323-r1      Rule-set: h323-nat
Rule-Id                      : 1
Rule position                 : 1
From zone                    : private
To zone                      : public
Match
  Source addresses           : 0.0.0.0      - 255.255.255.255
  Destination port           : 0            - 0
Action                        : interface
  Persistent NAT type        : N/A
  Persistent NAT mapping type : address-port-mapping
  Inactivity timeout         : 0
  Max session number         : 0
Translation hits              : 0
  Successful sessions        : 0
  Failed sessions            : 0
  Number of sessions         : 0
```

Meaning

The Translation hits field shows that, there is no traffic matching the source NAT rule.

Example: Using NAT with the H.323 ALG to Enable Outgoing Calls

IN THIS SECTION

- [Requirements | 223](#)
- [Overview | 223](#)
- [Configuration | 224](#)
- [Verification | 230](#)

This example shows how to configure static NAT with H.323 ALG to enable calls from a private to a public network.

Requirements

Before you begin, understand the H.323 ALG and its processes. See ["Understanding H.323 ALG" on page 192](#).

Overview

IN THIS SECTION

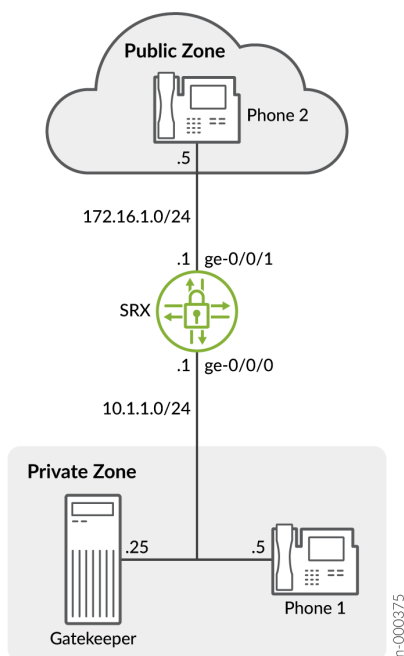
- [Topology | 223](#)

In this example (see [Figure 22 on page 224](#)), IP-Phone 1 and a server called gatekeeper are in the private zone and IP-Phone2 is in the public zone. You configure static NAT to enable IP-Phone1 and gatekeeper to call IP-Phone2 in the public zone. You then create a policy called public-to-private to allow ALG H.323 traffic from the public zone to the private zone and a policy called private-to-public to allow ALG H.323 traffic from the private zone to the public zone.

Topology

[Figure 22 on page 224](#) shows NAT with the H.323 ALG outgoing calls.

Figure 22: NAT with the H.323 ALG—Outgoing Calls



In this example, you configure static NAT as follows:

- Create a static NAT rule set called `ip-phones` with a rule called `phone1` to match packets from the public zone with the destination address `172.16.1.5/32`. For matching packets, the destination IP address is translated to the private address `10.1.1.5/32`.
- Define a second rule called `gatekeeper` to match packets from the public zone with the destination address `172.16.1.25/32`. For matching packets, the destination IP address is translated to the private address `10.1.1.25/32`.
- Create proxy ARP for the addresses `172.16.1.5/32` and `172.16.1.25/32` on interface `ge-0/0/1`. This allows the system to respond to ARP requests received on the specified interface for these addresses.

Configuration

IN THIS SECTION

- Procedure | [225](#)

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24
set security zones security-zone private address-book address IP-Phone1 10.1.1.5/32
set security zones security-zone private address-book address gatekeeper 10.1.1.25/32
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address IP-Phone2 172.16.1.5/32
set security zones security-zone public interfaces ge-0/0/1.0
set security nat static rule-set ip-phones from zone public
set security nat static rule-set ip-phones rule phone1 match destination-address 172.16.1.5/32
set security nat static rule-set ip-phones rule phone1 then static-nat prefix 10.1.1.5/32
set security nat static rule-set ip-phones rule gatekeeper match destination-address
172.16.1.25/32
set security nat static rule-set ip-phones rule gatekeeper then static-nat prefix 10.1.1.25/32
set security nat proxy-arp interface ge-0/0/1.0 address 172.16.1.5/32
set security nat proxy-arp interface ge-0/0/1.0 address 172.16.1.25/32
set security policies from-zone public to-zone private policy public-to-private match source-
address IP-Phone2
set security policies from-zone public to-zone private policy public-to-private match
destination-address gatekeeper
set security policies from-zone public to-zone private policy public-to-private match
application junos-h323
set security policies from-zone public to-zone private policy public-to-private then permit
set security policies from-zone private to-zone public policy private-to-public match source-
address IP-Phone1
set security policies from-zone private to-zone public policy private-to-public match source-
address gatekeeper
set security policies from-zone private to-zone public policy private-to-public match
destination-address IP-Phone2
set security policies from-zone private to-zone public policy private-to-public match
application junos-h323
set security policies from-zone private to-zone public policy private-to-public then permit
```

Step-by-Step Procedure

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

To configure static NAT with the H.323 ALG to enable calls from a private to a public network:

1. Configure interfaces.

```
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24
```

2. Create zones and assign addresses to them.

```
[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/1.0
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone private address-book address IP-Phone1 10.1.1.5/32
user@host# set security-zone private address-book address gatekeeper 10.1.1.25/32
user@host# set security-zone public interfaces ge-0/0/1.0
user@host# set security-zone public address-book address IP-Phone2 172.16.1.5/32
```

3. Configure static NAT rule set with rules.

```
[edit security nat static rule-set ip-phones]
user@host# set from zone public
user@host# set rule phone1 match destination-address 172.16.1.5/32
user@host# set rule phone1 then static-nat prefix 10.1.1.5/32
user@host# set rule gatekeeper match destination-address 172.16.1.25/32
user@host# set rule gatekeeper then static-nat prefix 10.1.1.25/32
```

4. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/1 address 172.16.1.5/32
user@host# set proxy-arp interface ge-0/0/1 address 172.16.1.25/32
```

5. Configure a security policy for incoming traffic.

```
[edit security policies from-zone public to-zone private policy public-to-private]
user@host# set match source-address IP-Phone2
user@host# set match destination-address gatekeeper
user@host# set match application junos-h323
user@host# set then permit
```

6. Configure a security policy for outgoing traffic.

```
[edit security policies from-zone private to-zone public policy private-to-public]
user@host# set match source-address IP-Phone1
user@host# set match source-address gatekeeper
user@host# set match destination-address IP-Phone2
user@host# set match application junos-h323
user@host# set then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, `show security nat`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 10.1.1.1/24;
      }
    }
  }
  ge-0/0/1 {
    unit 0 {
      family inet {
        address 172.16.1.1/24;
      }
    }
  }
```



```

    }
}
[edit]
user@host# show security zones
security-zone private {
    address-book {
        address IP-Phone1 10.1.1.5/32;
        address gatekeeper 10.1.1.25/32;
    }
    interfaces {
        ge-0/0/0.0;
    }
}

security-zone public {
    address-book {
        address IP-Phone2 172.16.1.5/32;
    }
    interfaces {
        ge-0/0/1.0;
    }
}
[edit]
user@host# show security nat
static {
    rule-set ip-phones {
        from zone public;
        rule phone1 {
            match {
                destination-address 172.16.1.5/32;
            }
            then {
                static-nat prefix 10.1.1.5/32;
            }
        }
        rule gatekeeper {
            match {
                destination-address 172.16.1.25/32;
            }
            then {
                static-nat prefix 10.1.1.25/32;
            }
        }
    }
}

```

```

    }
    proxy-arp {

        interface ge-0/0/1.0 {
            address {
                172.16.1.5/32;
                172.16.1.25/32;
            }
        }
    }
}
[edit]
user@host# show security policies
    from-zone public to-zone private {
        policy public-to-private {
            match {
                source-address IP-Phone2;
                destination-address gatekeeper;
                application junos-h323;
            }
            then {
                permit;
            }
        }
    }
    from-zone private to-zone public {
        policy private-to-public {
            match {
                source-address [ IP-Phone1 gatekeeper ];
                destination-address IP-Phone2;
                application junos-h323;
            }
            then {
                permit;
            }
        }
    }
}

```

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

Verification

IN THIS SECTION

- [Verifying H.323 ALG Status | 230](#)
- [Verifying Security ALG H.323 Counters | 231](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying H.323 ALG Status

Purpose

Verify that H.323 ALG is enabled on your system.

Action

From operational mode, enter the `show security alg status` command.

```
user@host> show security alg status
```

```
ALG Status :
```

```
DNS      : Enabled
FTP       : Enabled
H323     : Enabled
MGCP     : Enabled
MSRPC    : Enabled
PPTP     : Enabled
RSH      : Enabled
RTSP     : Enabled
SCCP     : Enabled
SIP      : Enabled
SQL      : Enabled
SUNRPC   : Enabled
TALK     : Enabled
TFTP     : Enabled
IKE-ESP  : Disabled
```

Meaning

The output shows the H323 ALG status as follows:

- Enabled—Shows the H323 ALG is enabled.
- Disabled—Shows the H323 ALG is disabled.

Verifying Security ALG H.323 Counters

Purpose

Verify that there is a security counters for ALG H323.

Action

From operational mode, enter the `show security alg h323 counters` command.

```
user@host> show security alg h323 counters
```

```
H.323 counters summary:  
Packets received :4060  
Packets dropped :24  
RAS message received :3690Q.931 message received :202  
H.245 message received :145  
Number of calls :25  
Number of active calls :0
```

```
H.323 Error Counters:  
Decoding errors :24  
Message flood dropped :0  
NAT errors :0  
Resource manager errors :0
```

```
H.323 Message Counters:  
RRQ : 431  
RCF : 49  
ARQ : 60  
ACF : 33  
URQ : 34  
UCF : 25
```

```

DRQ : 55
DCF : 44
oth RAS : 2942
Setup : 28
Alert : 9
Connect : 25
CallPrcd : 18
Info : 0
RelCmpl : 39
Facility : 14
Progress : 0
Empty : 65
OLC : 20
OLC-ACK : 20

```

Meaning

The sample output gives the synopsis of security ALG H.323 counters expressing that there are security counters for ALG H.323.

Example: Setting H.323 ALG Endpoint Registration Timeouts

IN THIS SECTION

- Requirements | 232
- Overview | 233
- Configuration | 233
- Verification | 234

This example shows how to specify the endpoint registration timeout.

Requirements

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

Overview

In Network Address Translation (NAT) mode, when endpoints in the protected network behind the Juniper Networks device register with the H.323 gatekeeper, the device adds an entry to the NAT table containing a mapping of the public-to-private address for each endpoint. These entries make it possible for endpoints in the protected network to receive incoming calls.

You set an endpoint registration timeout to specify how long an endpoint registration entry remains in the NAT table. To ensure uninterrupted incoming call service, set the endpoint registration timeout to a value equal to or greater than the keepalive value the administrator configures on the gatekeeper. The range is 10 to 50,000 seconds, the default value is 3600 seconds.

Configuration

IN THIS SECTION

- [Procedure | 233](#)

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To specify the H.323 ALG endpoint registration timeout:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. In the **Timeout for endpoints** box, type **5000**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

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

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

Verification

To verify the configuration is working properly, enter the `show security alg h323 counters` command.

Example: Setting H.323 ALG Media Source Port Ranges

IN THIS SECTION

- Requirements | 234
- Overview | 234
- Configuration | 235
- Verification | 236

This example shows how to enable the H.323 ALG media source port feature.

Requirements

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

Overview

The media source port feature enables you to configure the device to allow media traffic on a narrow or wide range of ports. By default, the device listens for H.323 traffic on a narrow range of ports. If your endpoint equipment allows you to specify a sending port and a listening port, you might want to narrow the range of ports the device allows media traffic on. This enhances security by opening a smaller pinhole for H.323 traffic. This example shows how to configure the device to open a wide gate for media traffic by enabling the media source port feature.

Configuration

IN THIS SECTION

- [Procedure](#) | 235

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To enable the H.323 ALG media source port feature:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. Select the **Enable Permit media from any source port** check box.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To enable the H.323 ALG media source port feature:

1. Set a narrow gate for media traffic by disabling the media source port for the H.323 ALG.

```
[edit]  
user@host# delete security alg h323 media-source-port-any
```

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

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


Verification

To verify the configuration is working properly, enter the `show security alg h323 counters` command.

Example: Configuring H.323 ALG DoS Attack Protection

IN THIS SECTION

- Requirements | 236
- Overview | 236
- Configuration | 236
- Verification | 237

This example shows how to configure the H.323 ALG DoS attack protection feature.

Requirements

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

Overview

You can protect the H.323 gatekeeper from denial-of-service (DoS) flood attacks by limiting the number of Registration, Admission, and Status (RAS) messages per second it will attempt to process. Incoming RAS request messages exceeding the threshold you specify are dropped by H.323 Application Layer Gateway (ALG). The range is 2 to 50,000 messages per second, the default value is 1000.

Configuration

IN THIS SECTION

- Procedure | 237

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To configure the H.323 ALG DoS attack protection feature:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. In the **Message flood gatekeeper threshold** box, type **5000**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure the H.323 ALG DoS attack protection feature:

1. Configure the gatekeeper for the H.323 ALG and set the threshold.

```
[edit]  
user@host# set security alg h323 application-screen message-flood gatekeeper threshold 5000
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg h323 counters` command.

Understanding H.323 ALG Known Message Types

IN THIS SECTION

- [H.225 RAS Signaling: Gatekeepers and Gateways | 238](#)
- [H.225 Call Signaling \(Q.931\) | 243](#)
- [H.245 Media Control and Transport signaling | 243](#)

The H.323 standard is a legacy voice-over-IP (VoIP) protocol defined by the International Telecommunication Union (ITU-T). H.323 consists of a suite of protocols (such as H.225.0 and H.245) that are used for call signaling and call control for VoIP. There are three major processes in H.323:

- **Gatekeeper Discovery**—An endpoint finds its gatekeeper through the gatekeeper discovery process, through broadcast or unicast (to a known IP and the well-known UDP port 1719).
- **Endpoint Registration, Admission, and Status**—An endpoint registers to a gatekeeper and asks for its management. Before making a call, an endpoint asks its gatekeeper for permission to place the call. In both registration and admission phases, the Registration, Admission, and Status (RAS) channel is used.
- **Call Control and Call Setup**—Calls can be established within a zone or across two zones, or even across multiple zones (multipoint conference). The call setup and tear down is performed through the call signaling channel whose TSAP is the well-known TCP port (1720). The call control, including opening/closing media channels between two endpoints, is performed through the call control channel whose TSAP is dynamically assigned from the previous call signaling process. H.245 messages are used in the call control channel, and are encoded using ASN.1.

H.225 RAS Signaling: Gatekeepers and Gateways

Registration, Admission, and Status (RAS), as described in the (ITU-T) H.323 standard, is the signaling protocol used between gateways or endpoints. The gatekeepers provide address resolution and admission control services.

RAS is the process by which H.323 gateways discover their zone gatekeepers. RAS communication is carried out through a UDP datagram on port 1718 (multicast) and 1719 (unicast). Endpoints use the RAS protocol to communicate with a gatekeeper. If an H.323 endpoint does not know its gatekeeper, then it can send a Gatekeeper Request (GRQ) message to seek the gatekeeper's response. One or more gatekeepers might answer the request with either a Gatekeeper Confirmation (GCF) message or a Gatekeeper Reject (GRJ) message. A reject message contains the reason for rejection.

[Table 6 on page 239](#) lists the supported RAS gatekeeper messages.

Table 6: Gatekeeper Messages

Message	Description
GRQ (Gatekeeper_Request)	A message sent from an endpoint to a gatekeeper to "discover" gatekeepers willing to provide service.
GCF (Gatekeeper_Confirm)	A reply from a gatekeeper to an endpoint that indicates the acceptance to communicate with the gatekeeper's RAS channel.
GRJ (Gatekeeper_Reject)	A reply from a gatekeeper to an endpoint that rejects the endpoint request.

RAS Registration and Unregistration

Registration is the process by which the gateways, terminals, and multipoint control units (MCUs) join a zone and inform the gatekeeper of their IP and alias addresses. Every gateway can register only one active gatekeeper.

The registration takes place after the endpoint determines and confirms the gatekeeper to communicate, by sending a Registration Request (RRQ) message. The gatekeeper then responds with a Registration Confirm (RCF) message, thereby making the endpoint known to the network.

[Table 7 on page 239](#) lists the supported RAS registration and unregistration messages.

Table 7: Registration and Unregistration Messages

Message	Description
RRQ (Registration_Request)	A message sent from an endpoint to a gatekeeper. Registration requests are predefined in the system's administrative setup.
RCF (Registration_Confirm)	A reply from a gatekeeper that confirms an endpoint's registration in response to an RRQ message.
RRJ (Registration_Reject)	A reply from a gatekeeper that rejects an endpoint's registration.

Table 7: Registration and Unregistration Messages *(Continued)*

Message	Description
URQ (Unregister_Request)	A message sent from an endpoint or a gatekeeper requesting to cancel a registration.
UCF (Unregister_Confirm)	A reply sent from an endpoint or a gatekeeper to confirm that the registration is canceled.
URJ (Unregister_Reject)	A message that indicates that the endpoint is not preregistered with the gatekeeper.

RAS Admissions

Admission messages between endpoints and gatekeepers provide the basis for call admissions and bandwidth control. The gatekeeper then resolves the address either with confirmation or rejection of an admission request.

[Table 8 on page 240](#) lists the supported RAS admission messages.

Table 8: Call Admission Messages

Message	Description
ARQ (Admission_Request)	An attempt by an endpoint to initiate a call.
ACF (Admission_Confirm)	A positive response from a gatekeeper that authorizes an endpoint to participate in a call.
ARJ (Admission_Reject)	A message sent from a gatekeeper rejecting the ARQ message that initiates a call.

RAS Location

Location Request (LRQ) messages are sent by either an endpoint or a gatekeeper to an interzone gatekeeper to get the IP addresses of different zone endpoints.

[Table 9 on page 241](#) lists the supported RAS location request messages.

Table 9: Location Request Messages

Message	Description
LRQ (Location_Request)	A message sent to request a gatekeeper for contact information of one or more addresses.
LCF (Location_Confirm)	A response sent by a gatekeeper that contains call signaling channel or RAS channel addresses.
LRJ (Location_Reject)	A response sent by gatekeepers that received an LRQ for which the requested endpoint is not registered.

RAS Bandwidth Control

Bandwidth control is invoked to set up the call, and is initially managed through the admission messages (ARQ/ACF/ARJ) sequence.

[Table 10 on page 241](#) lists the supported RAS bandwidth control messages.

Table 10: Bandwidth Control Messages

Message	Description
BRQ (Bandwidth_Request)	A request sent by an endpoint to a gatekeeper to increase or decrease call bandwidth.
BCF (Bandwidth_Confirm)	A response sent by a gatekeeper to confirm the acceptance of a bandwidth change request.
BRJ (Bandwidth_Reject)	A response sent by a gatekeeper to reject a bandwidth change request.

RAS Status Information

A gatekeeper uses an Information Request (IRQ) message to determine the status of an endpoint. The RAS protocol is used to determine whether the endpoints are online or offline.

[Table 11 on page 242](#) lists the supported RAS status information messages.

Table 11: Status Information Messages

Message	Description
IRQ (Information_Request)	A message sent from a gatekeeper to request status information of its recipient endpoints.
IRR (Information_Request_Response)	A response sent by endpoint to a gatekeeper in response to an IRQ message. It determines whether the endpoints are online or offline.
IACK (Info_Request_Acknowledge)	A message sent by a gatekeeper to acknowledge the receipt of an IRR message from an endpoint.
INACK (Info_Request_Neg_Acknowledge)	A message sent a gatekeeper if an information request message is not understood.

RAS Disengage Information

An endpoint sends a Disengage Request (DRQ) message to a gatekeeper in the event of a call drop.

[Table 12 on page 242](#) lists the supported RAS disengage messages.

Table 12: Disengage Request Messages

Message	Description
DRQ (Disengage_Request)	A status request sent from an endpoint to a gatekeeper when a call ends.
DCF (Disengage_Confirm)	A message sent by a gatekeeper to confirm receipt of the DRQ message from an endpoint.
DRJ (Disengage_Reject)	A message sent by a gatekeeper that rejects a disengage confirmation request from an endpoint.

H.225 Call Signaling (Q.931)

H.225 is used to set up connections between H.323 endpoints. The (ITU-T) H.225 recommendation specifies the use and support of Q.931 messages.

H.225 call signaling supports the following messages:

- Setup and Setup Acknowledge
- Call Proceeding
- Connect
- Alerting
- User Information
- Release Complete
- Facility
- Progress
- Status and Status Inquiry
- Notify

H.245 Media Control and Transport signaling

H.245 handles end-to-end control messages between H.323 endpoints. This control channel protocol establishes the logical channels for transmission of audio, video, data, and control channel information.

H.245 supports the following messages:

- Request
- Response
- Command
- Indication

Understanding H.323 ALG Unknown Message Types

Unknown H.323 message type feature enables you to specify how unidentified H.323 messages are handled by the device. The default is to drop unknown (unsupported) messages.

You can protect the H.323 gatekeeper from denial-of-service (DoS) flood attacks by limiting the number of Registration, Admission, and Status (RAS) messages per second it will attempt to process. Incoming RAS request messages exceeding the threshold you specify are dropped by the H.323 Application Layer Gateway (ALG). The range is 2 to 50,000 messages per second, the default value is 1000.

We do not recommend permitting unknown messages because they can compromise security. However, in a secure test or production environment, unknown H.323 message type command can be useful for resolving interoperability issues with disparate vendor equipment. Permitting unknown H.323 messages can help you get your network operational, so that you can analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped. The unknown H.323 message type feature enables you to configure the device to accept H.323 traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.



NOTE: Unknown H.323 message type option applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

Example: Allowing Unknown H.323 ALG Message Types

IN THIS SECTION

- Requirements | 244
- Overview | 245
- Configuration | 245
- Verification | 246

This example shows how to configure the device to allow unknown H.323 message types in both route and NAT modes.

Requirements

Before you begin, understand and configure any Avaya H.323-specific features. See the *Administrator Guide for Avaya Communication Manager*, *Avaya IP Telephony Implementation Guide*, and *Avaya Application Solutions IP Telephony Deployment Guide* at <http://support.avaya.com>.

Overview

This feature enables you to specify how unidentified H.323 messages are handled by the device. The default is to drop unknown (unsupported) messages. The Enable Permit NAT applied option and the permit-nat-applied configuration statement specify that unknown messages be allowed to pass if the session is in NAT mode. The Enable Permit routed option and the permit-routed configuration statement specify that unknown messages be allowed to pass if the session is in route mode. (Sessions in transparent mode are treated as route mode.)

Configuration

IN THIS SECTION

- [Procedure | 245](#)

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To configure the device to allow unknown H.323 message types in both route and NAT modes:

1. Select **Configure>Security>ALG**.
2. Select the **H323** tab.
3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure the device to allow unknown H.323 message types in both route and NAT modes:

- 1. Specify that unknown messages be allowed to pass if the session is in NAT mode.

```
[edit]
user@host# set security alg h323 application-screen unknown-message permit-nat-applied
```

- 2. Specify that unknown messages be allowed to pass if the session is in route mode.

```
[edit]
user@host# set security alg h323 application-screen unknown-message permit-routed
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg h323` command and the `show security alg h323 counters` command.

Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
17.4R1	Starting with Junos OS Release 17.4R1, the gateway-to-gateway call feature is supported on the H.323 Application Layer Gateway (ALG).
17.4R1	Starting with Junos OS Release 17.4R1, the H.323 Application Layer Gateway (ALG) supports NAT64 rules in an IPv6 network.

RELATED DOCUMENTATION

Understanding VoIP ALG Types 188
SCCP ALG 294
SIP ALG 326

MGCP ALG

IN THIS SECTION

- [Understanding the MGCP ALG | 247](#)
- [MGCP ALG Configuration Overview | 254](#)
- [Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs | 255](#)
- [Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALG and NAT | 265](#)
- [Understanding MGCP ALG Call Duration and Timeouts | 283](#)
- [Example: Setting MGCP ALG Call Duration | 284](#)
- [Example: Setting MGCP ALG Inactive Media Timeout | 286](#)
- [Example: Setting MGCP ALG Transaction Timeout | 288](#)
- [Example: Configuring MGCP ALG DoS Attack Protection | 290](#)
- [Example: Allowing Unknown MGCP ALG Message Types | 292](#)

The Media Gateway Control Protocol (MGCP) is a text-based signaling and call control communications protocol used in VoIP telecommunication systems. MGCP is used to set up, maintain, and terminate calls between multiple endpoints.

Understanding the MGCP ALG

IN THIS SECTION

- [MGCP Security | 248](#)
- [Entities in MGCP | 249](#)
- [MGCP Commands | 250](#)
- [MGCP Response Codes | 253](#)

The Media Gateway Control Protocol (MGCP) is a text-based Application Layer protocol used for call setup and call control between the media gateway and the media gateway controller (MGC).

The protocol is based on a primary/client call control architecture: the MGC (call agent) maintains call control intelligence, and media gateways carry out the instructions from the call agent. Both signaling packets and media packets are transmitted over UDP. Junos OS supports MGCP in route mode and Network Address Translation (NAT) mode.

The MGCP Application Layer Gateway (ALG) performs the following procedures:

- Conducts voice-over-IP (VoIP) signaling payload inspection. The payload of the incoming VoIP signaling packet is fully inspected based on related RFCs and proprietary standards. Any malformed packet attack is blocked by the ALG.
- Conducts MGCP signaling payload inspection. The payload of the incoming MGCP signaling packet is fully inspected in accordance with RFC 3435. Any malformed-packet attack is blocked by the ALG.
- Provides stateful processing. The corresponding VoIP-based state machines are invoked to process the parsed information. Any out-of-state or out-of-transaction packet is identified and properly handled.
- Performs NAT. Any embedded IP address and port information in the payload is properly translated based on the existing routing information and network topology, and is then replaced with the translated IP address and port number, if necessary.
- Manages pinholes for VoIP traffic. To keep the VoIP network secure, the IP address and port information used for media or signaling is identified by the ALG, and any needed pinhole is dynamically created and closed during call setup.

This topic contains the following sections:

MGCP Security

The MGCP ALG includes the following security features:

- Denial-of-service (DoS) attack protection. The ALG performs stateful inspection at the UDP packet level, the transaction level, and the call level. MGCP packets matching the RFC 3435 message format, transaction state, and call state, are processed. All other messages are dropped.
- Security policy enforcement between gateway and gateway controller (signaling policy).
- Security policy enforcement between gateways (media policy).
- Per-gateway MGCP message flooding control. Any malfunctioning or hacked gateway will not disrupt the whole VoIP network. Combined with per-gateway flooding control, damage is contained within the impacted gateway.

- Per-gateway MGCP connection flooding control.
- Seamless switchover/failover if calls, including calls in progress, are switched to the standby firewall in case of system failure.

Entities in MGCP

There are four basic entities in MGCP:

Endpoint

A media gateway is a collection of endpoints. An endpoint can be an analog line, trunk, or any other access point. An endpoint contains the following elements:

```
local-endpoint-name@domain-name
```

The following examples are some valid endpoint IDs:

```
group1/Trk8@example.net  
group2/Trk1/*@[192.168.10.8] (wild-carding)  
$@example.net (any endpoint within the media gateway)  
*@example.net (all endpoints within the media gateway)
```

Connection

Connections are created on each endpoint by an MG during call setup. A typical VoIP call involves two connections. A complex call, for example a three-party call or conference call, might require more connections. The MGC can instruct media gateways to create, modify, delete, and audit a connection.

A connection is identified by its connection ID, which is created by the MG when it is requested to create a connection. Connection ID is presented as a hexadecimal string, and its maximum length is 32 characters.

Call

A call is identified by its call ID, which is created by the MGC when establishing a new call. Call ID is a hexadecimal string with a maximum length of 32 characters. Call ID is unique within the MGC. Two or more connections can have the same call ID if they belong to the same call.

Call Agent

One or more call agents (also called media gateway controllers) are supported in MGCP to enhance reliability in the VoIP network. The following two examples are of call agent names:

```

CallAgent@voipCA.example.com
voipCA.example.com

```

Several network addresses can be associated under one domain name in the Domain Name System (DNS). By keeping track of the time to live (TTL) of DNS query/response data and implementing retransmission using other alternative network addresses, switchover and failover is achieved in MGCP.

The concept of a *notified entity* is essential in MGCP. The notified entity for an endpoint is the call agent currently controlling that endpoint. An endpoint should send any MGCP command to its notified entity. However, different call agents might send MGCP commands to this endpoint.

The notified entity is set to a provisioned value upon startup, but can be changed by a call agent through the use of the `NotifiedEntity` parameter contained in an MGCP message. If the notified entity for an endpoint is empty or has not been set explicitly, its value defaults to the source address of the last successful non-audit MGCP command received for that endpoint.

MGCP Commands

The MGCP protocol defines nine commands for controlling endpoints and connections. All commands are composed of a command header, optionally followed by Session Description Protocol (SDP) information. A command header has the following elements:

- A command line: command verb + transaction ID + endpointId + MGCP version.
- Zero or more parameter lines, composed of a parameter name followed by a parameter value.

[Table 13 on page 251](#) lists supported MGCP commands and includes a description of each, the command syntax, and examples. Refer to RFC 2234 for a complete explanation of command syntax.

Table 13: MGCP Commands

Command	Description	Command Syntax	Example
EPCF	EndpointConfiguration—Used by a call agent to inform a gateway of coding characteristics (a-law or mu-law) expected by the line side of the endpoint.	ReturnCode [PackageList] EndpointConfiguration (EndpointId, [BearerInformation])	EPCF 2012 wxx/T2@example.com MGCP 1.0B: e:mu
CRCX	CreateConnection—Used by a call agent to instruct the gateway to create a connection with, and endpoint inside, the gateway.	ReturnCode, [ConnectionId,] [SpecificEndPointId,] [LocalConnectionDescriptor,] [SecondEndPointId,] [SecondConnectionId,] [PackageList] CreateConnection (CallId, EndpointId, [NotifiedEntity,] [LocalConnectionOption,] Mode, [{RemoteConnectionDescriptor SecondEndpointId},] [encapsulated RQNT,] [encapsulated EPCF])	CRCX 1205 aaln/ 1@gw-25.example.net MGCP 1.0C: A3C47F21456789F0L: p:10, a:PCMUM: sendrecvX: 0123456789ADR: L/hdS: L/ rgv=0o=- 25678 753849 IN IP4 128.96.41.1s=-c=IN IP4 128.96.41.1t=0 0m=audio 3456 RTP/AVP 0
MDCX	ModifyConnection—Used by a call agent to instruct a gateway to change the parameters for an existing connection.	ReturnCode, [LocalConnectionDescriptor,] [PackageList] ModifyConnection (CallId, EndpointId, ConnectionId, [NotifiedEntity,] [LocalConnectionOption,] [Mode,] [RemoteConnectionDescriptor,] [encapsulated RQNT,] [encapsulated EPCF])	MDCX 1210 aaln/ 1@rgw-25.example.net MGCP 1.0C: A3C47F21456789F0I: FDE234C8M: recvonlyX: 0123456789AER: L/ huS: G/rtv=0o=- 4723891 7428910 IN IP4 128.96.63.25s=-c=IN IP4 128.96.63.25t=0 0m=audio 3456 RTP/AVP 0

Table 13: MGCP Commands (*Continued*)

Command	Description	Command Syntax	Example
DLCX	<p>DeleteConnection—Used by a call agent to instruct a gateway to delete an existing connection.</p> <p>DeleteConnection can also be used by a gateway to release a connection that can no longer be sustained.</p>	ReturnCode, ConnectionParameters, [PackageList] DeleteConnection (CallId, EndpointId, ConnectionId, [NotifiedEntity,] [encapsulated RQNT,] [encapsulated EPCF])	<p>Example 1: MGC -> MG</p> <pre>DLCX 9210 aaln/ 1@rgw-25.example.net MGCP 1.0C: A3C47F21456789F0I: FDE234C8</pre> <p>Example 2: MG -> MGC</p> <pre>DLCX 9310 aaln/ 1@rgw-25.example.net MGCP 1.0C: A3C47F21456789F0I: FDE234C8E: 900 - Hardware errorP: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48</pre>
RQNT	NotificationRequest command—Used by a call agent to instruct an MG to monitor for certain event(s) or signal(s) for a specific endpoint.	ReturnCode, [PackageList] NotificationRequest[(EndpointId, [NotifiedEntity,] [RequestedEvents,] RequestIdentifier, [DigitMap,] [SignalRequests,] [QuarantineHandling,] [DetectEvents,] [encapsulated EPCF])	<pre>RQNT 1205 aaln/ 1@rgw-25.example.net MGCP 1.0N: ca-new@callagent- ca.example.netX: 0123456789AAR: L/hd(A, E(S(L/dl),R(L/oc,L/ hu,D/[0-9#*T](D)))D: (0T 00T xx 91xxxxxxxxxx 9011x.T)S:T: G/ft</pre>
NTFY	Notify—Used by a gateway to inform the call agent when requested event(s) or signal(s) occur.	ReturnCode, [PackageList] Notify (EndpointID, [NotifiedEntity,] RequestIdentifier, ObservedEvents)	<pre>NTFY 2002 aaln/ 1@rgw-25.example.net MGCP 1.0N: ca@ca1.example.net:5678X: 0123456789ACO: L/ hd,D/9,D/1,D/2,D/0,D/1,D/8,D/2, D/9,D/4, D/2,D/6,D/6</pre>

Table 13: MGCP Commands (*Continued*)

Command	Description	Command Syntax	Example
AUEP	AuditEndpoint—Used by a call agent to audit the status of the endpoint.	ReturnCode, EndPointIdList, { [RequestedEvents,] [QuarantineHandling,] [DigitMap,] [SignalRequests,] [RequestedIdentifier,] [NotifiedEntity,] [ConnectionIdentifier,] [DetectEvents,] [ObservedEvents,] [EventStats,] [BearerInformation,] [BearerMethod,] [RestartDelay,] [ReasonCode,] [MaxMGCPDatagram,] [Capabilities]} [PackageList] AuditEndpoint (EndpointId, [RequestedInfo])	Example 1: AUEP 1201 aaIn/ 1@rgw-25.example.net MGCP 1.0F: A, R,D,S,X,N,I,T,O Example 2: AUEP 1200 *@rgw-25.example.net MGCP 1.0
AUCX	AuditConnection—Used by a call agent to collect the parameters applied to a connection.	ReturnCode, [CallId,] [NotifiedEntity,] [LocalConnectionOptions,] [Mode,] [RemoteConnectionDescriptor,] [LocalConnectionDescriptor,] [ConnectionParameters,] [PackageList] AuditConnection (EndpointId, ConnectionId, RequestedInfo)	AUCX 3003 aaIn/ 1@rgw-25.example.net MGCP 1.0I: 32F345E2F: C,N,L,M,LC,P
RSIP	RestartInProgress—Used by a gateway to notify a call agent that one or more endpoints are being taken out of service or placed back in service.	ReturnCode, [NotifiedEntity,] [PackageList] RestartInProgress (EndpointId, RestartMethod, [RestartDelay,] [ReasonCode])	RSIP 5200 aaIn/ 1@rg2-25.example.net MGCP 1.0RM: gracefulRD: 300

MGCP Response Codes

Every command sent by the calling agent or gateway, whether successful or not, requires a response code. The response code is in the header of the response message, and optionally is followed by session description information.

The response header is composed of a response line, followed by zero or more parameter lines, each containing a parameter name letter followed by its value. The response header is composed of a three-digit response code, transaction ID, and optionally followed by commentary. The response header in the following response message shows response code 200 (successful completion), followed by ID 1204 and the comment:OK.

```
200 1204 OK
I: FDE234C8
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 96
a=rtpmap:96 G726-32/8000
```

The ranges of response codes are defined as follows:

- 000 – 099 indicate a response acknowledgement.
- 100 – 199—indicate a provisional response.
- 200 – 299 indicate a successful completion (final response).
- 400 – 499 indicate a transient error (final response).
- 500 – 599 indicate a permanent error (final response).

Refer to RFC 3661 for detailed information about response codes.

A response to a command is sent to the source address of the command, not to the current notified entity. A media gateway can receive MGCP commands from various network addresses simultaneously, and send back responses to corresponding network addresses. However, it sends all MGCP commands to its current notified entity.

MGCP ALG Configuration Overview

The Media Gateway Control Protocol (MGCP ALG) is enabled by default on the device—no action is required to enable it. However, you might choose to fine-tune MGCP ALG operations by using the following instructions:

1. Free up bandwidth when calls fail to properly terminate. See ["Example: Setting MGCP ALG Call Duration" on page 284](#).

2. Control how long a call can remain active without any media traffic. See ["Example: Setting MGCP ALG Inactive Media Timeout" on page 286](#).
3. Track and clear signaling traffic when it times out. See ["Example: Setting MGCP ALG Transaction Timeout" on page 288](#).
4. Protect the media gateway from denial-of-service (DoS) flood attacks. See ["Example: Configuring MGCP ALG DoS Attack Protection" on page 290](#).
5. Enable unknown messages to pass when the session is in Network Address Translation (NAT) mode and route mode. See ["Example: Allowing Unknown MGCP ALG Message Types" on page 292](#).

Example: Configuring Media Gateways in Subscriber Homes Using MGCP ALGs

IN THIS SECTION

- [Requirements | 255](#)
- [Overview | 255](#)
- [Configuration | 257](#)
- [Verification | 261](#)

This example shows how to configure media gateways in subscriber homes using MGCP ALGs.

Requirements

Before you begin:

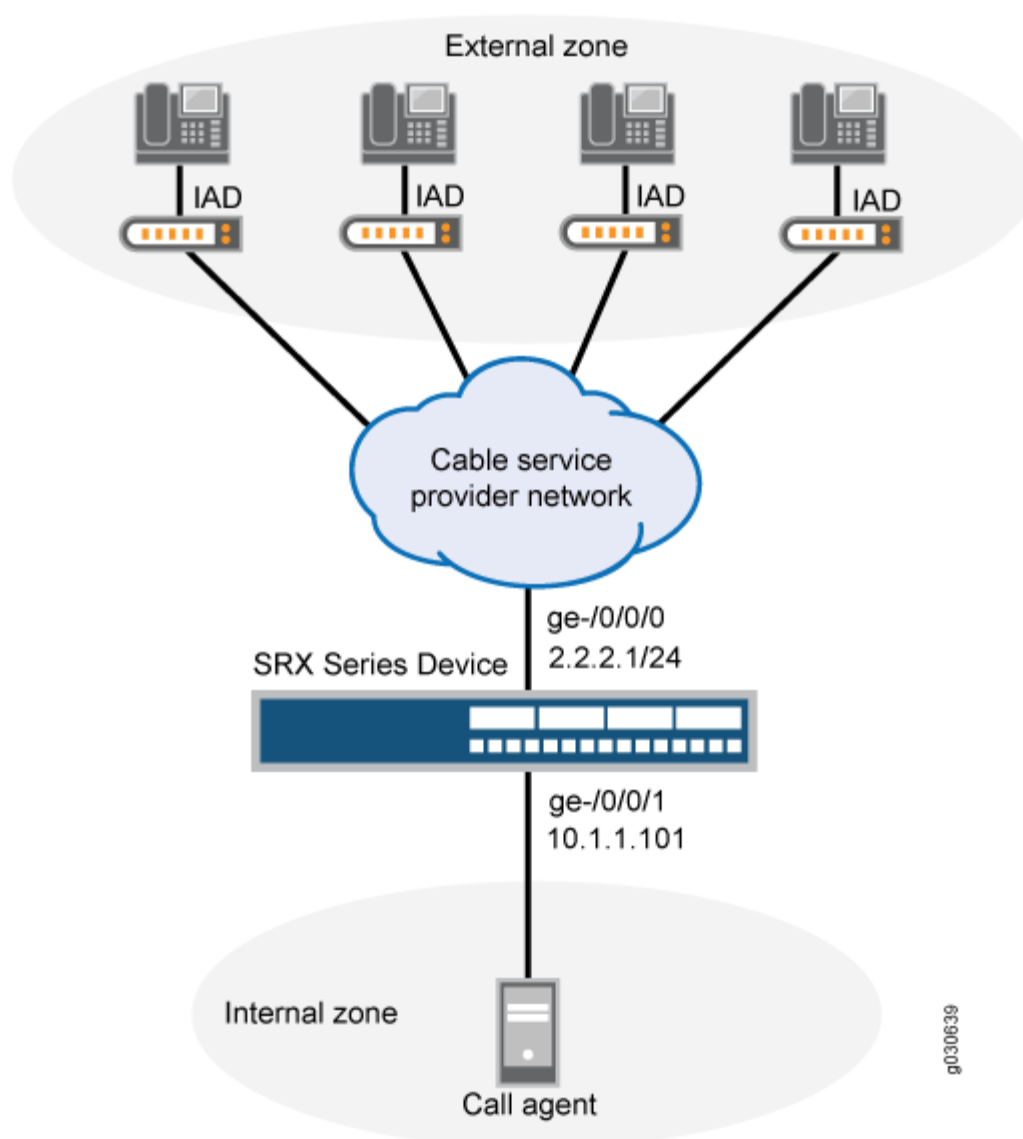
- Configure zones. See *Example: Creating Security Zones*.
- Configure addresses and interfaces. See *Example: Configuring Address Books and Address Sets*.
- Configure security policies. See *Security Policies Configuration Overview*.

Overview

When a cable service provider offers MGCP services to residential subscribers, they locate the Juniper Networks device and call agent on their premises and install a set-top box, in each subscriber's home. The set-top boxes act as gateways for the residences.

After creating zones—external_subscriber for the customer and internal_ca for the service provider—you configure addresses, then interfaces, and finally policies to allow signaling between endpoints. Note that although gateways frequently reside in different zones, requiring policies for media traffic, in this example both gateways are in the same subnet. Note also that because RTP traffic between the gateways never passes through the device, no policy is needed for the media. See [Figure 23 on page 256](#).

Figure 23: Media Gateway in Subscriber Homes



Configuration

IN THIS SECTION

- [Procedure | 257](#)

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security zones security-zone external-subscriber host-inbound-traffic system-services all
set security zones security-zone external-subscriber host-inbound-traffic protocols all
set security zones security-zone internal-ca host-inbound-traffic system-services all
set security zones security-zone internal-ca host-inbound-traffic protocols all
set interfaces ge-0/0/1 unit 0 family inet address 2.2.2.1/24
set interfaces ge-0/0/0 unit 0 family inet
set security zones security-zone external-subscriber interfaces ge-0/0/0
set security zones security-zone internal-ca interfaces ge-0/0/1
set security address-book book1 address ca-agent 110.1.1.101/32
set security address-book book1 attach zone internal-ca
set security address-book book2 address subscriber-subnet 2.2.2.1/24
set security address-book book2 attach zone external-subscriber
set security policies from-zone internal-ca to-zone external-subscriber policy ca-to-subscribers
match source-address ca-agent-1
set security policies from-zone internal-ca to-zone external-subscriber policy ca-to-subscribers
match destination-address subscriber-subnet
set security policies from-zone internal-ca to-zone external-subscriber policy ca-to-subscribers
match application junos-mgcp
set security policies from-zone internal-ca to-zone external-subscriber policy ca-to-subscribers
then permit
set security policies from-zone external-subscriber to-zone internal-ca policy subscriber-to-ca
match source-address subscriber-subnet
set security policies from-zone external-subscriber to-zone internal-ca policy subscriber-to-ca
match destination-address ca-agent-1
```

```

set security policies from-zone external-subscriber to-zone internal-ca policy subscriber-to-ca
match application junos-mgcp
set security policies from-zone external-subscriber to-zone internal-ca policy subscriber-to-ca
then permit
set security policies from-zone internal-ca to-zone internal-ca policy intra-ca then permit
set security policies from-zone external-subscriber to-zone external-subscriber policy intra-
subscriber match destination-address any

```

Step-by-Step Procedure

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

To configure media gateways in subscriber homes using MGCP ALGs:

1. Create security zones for the customer and the service provider.

```

[edit security zones security-zone external-subscriber]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all
[edit security zones security-zone internal-ca]
user@host# set host-inbound-traffic system-services all
user@host# set host-inbound-traffic protocols all

```

2. Configure interfaces for the zones.

```

[edit]
user@host# edit security zones security-zone external-subscriber interfaces ge-0/0/0
user@host# set interfaces ge-0/0/0 unit 0 family inet
user@host# set security zones security-zone internal-ca interfaces ge-0/0/1
user@host# set interfaces ge-0/0/1 unit 0 family inet address 2.2.2.1/24

```

3. Configure address books and attach zones to them.

```
[edit security address-book book1]
user@host# set address ca-agent 110.1.1.101/32
user@host# set attach zone internal-ca
```

```
[edit security address-book book2]
user@host# set address subscriber-subnet 2.2.2.1/24
user@host# set attach zone external-subscriber
```

4. Configure policies for traffic from the internal to the external zone.

```
[edit security policies from-zone internal-ca to-zone external-subscriber policy ca-to-
subscribers]
user@host# edit match source-address ca-agent-1
user@host# set match destination-address subscriber-subnet
user@host# set match application junos-mgcp
user@host# set then permit
```

5. Configure policies for traffic from the external to the internal zone.

```
[edit security policies from-zone external-subscriber to-zone internal-ca policy subscriber-
to-ca]
user@host# edit match source-address subscriber-subnet
user@host# set match destination-address ca-agent-1
user@host# set match application junos-mgcp
user@host# set then permit
```

6. Configure policies for traffic between two internal zones.

```
[edit security policies from-zone internal-ca to-zone internal-ca policy intra-ca]
user@host# edit match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set then permit
```


7. Configure policies for traffic between two external zones.

```
[edit security policies from-zone external-subscriber to-zone external-subscriber policy
intra-subscriber]
user@host# edit match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set then permit
```

Results

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

```
from-zone internal-ca to-zone external-subscriber {
  policy ca-to-subscribers {
    match {
      source-address ca-agent-1;
      destination-address subscriber-subnet;
      application junos-mgcp;
    }
    then {
      permit;
    }
  }
}
from-zone external-subscriber to-zone internal-ca {
  policy subscriber-to-ca {
    match {
      source-address subscriber-subnet;
      destination-address ca-agent-1;
      application junos-mgcp;
    }
    then {
      permit;
    }
  }
}
from-zone internal-ca to-zone internal-ca {
  policy intra-ca {
```

```
        match {
            ssource-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone external-subscriber to-zone external-subscriber {
    policy intra-subscriber {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
```

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

Verification

IN THIS SECTION

- [Verifying MGCP ALGs | 262](#)
- [Verifying MGCP ALG Calls | 262](#)
- [Verifying MGCP ALG Endpoints | 263](#)
- [Verifying MGCP ALG Counters | 264](#)

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

Verifying MGCP ALGs

Purpose

Verify the MGCP ALG verification options.

Action

From operational mode, enter the `show security alg mgcp ?` command.

```
user@host> show security alg mgcp ?
Possible completions:
  calls           Show MGCP calls
  counters        Show MGCP counters
  endpoints       Show MGCP endpoints
```

Meaning

The output shows a list of all MGCP verification parameters. Verify the following information:

- All MGCP calls
- Counters for all MGCP calls
- Information about all MGCP endpoints

Verifying MGCP ALG Calls

Purpose

Verify information about active MGCP calls.

Action

From operational mode, enter the `show security alg mgcp calls` command.

```
user@host> show security alg mgcp calls
Endpoint@GW           Zone      Call ID           RM Group
d001@101.50.10.1      Trust     10d55b81140e0f76  512
  Connection Id> 0
  Local SDP>  o: 101.50.10.1      x_o: 101.50.10.1
```

```

c: 101.50.10.1/32206      x_c: 101.50.10.1/32206
Remote SDP> c: 3.3.3.5/16928      x_c: 3.3.3.5/16928
Endpoint@GW              Zone      Call ID      RM Group
d001@3.3.3.5             Untrust    3a104e9b41a7c4c9    511
Connection Id> 0
Local SDP> o: 3.3.3.5          x_o: 3.3.3.5
c: 3.3.3.5/16928          x_c: 3.3.3.5/16928
Remote SDP> c: 101.50.10.1/32206      x_c: 101.50.10.1/32206

```

Meaning

The output displays information about all MGCP calls. Verify the following information:

- Endpoint
- Zone
- Call identifier
- Resource Manager group

Verifying MGCP ALG Endpoints

Purpose

Verify information about MGCP endpoints.

Action

From operational mode, enter the `show security alg mgcp endpoints` command.

```

user@host> show security alg mgcp endpoints
Gateway: 101.50.10.1 Zone: Trust IP: 101.50.10.1 -> 101.50.10.1
  Endpoint      Trans #  Call #  Notified Entity
  d001          1        1      0.0.0.0/0->0.0.0.0/0
Gateway: 3.3.3.5 Zone: Untrust IP: 3.3.3.5 -> 3.3.3.5
  Endpoint      Trans #  Call #  Notified Entity
  d001          1        1      0.0.0.0/0->0.0.0.0/0

```

Meaning

The output displays information about all MGCP endpoints. Verify the following information:

- Gateway IP address and zone of both endpoints
- Endpoint identifier, transaction number, call number, and notified entity for each gateway

Verifying MGCP ALG Counters

Purpose

Verify information about MGCP counters.

Action

From operational mode, enter the `show security alg mgcp counters` command.

```
user@host> show security alg mgcp counters
```

MGCP counters summary:

Packets received :284

Packets dropped :0

Message received :284

Number of connections :4

Number of active connections :3

Number of calls :4

Number of active calls :3

Number of transactions :121

Number of active transactions:52

Number of re-transmission :68

MGCP Error Counters:

Unknown-method :0

Decoding error :0

Transaction error :0

Call error :0

Connection error :0

Connection flood drop :0

Message flood drop :0

IP resolve error :0

NAT error :0

Resource manager error :0

MGCP Packet Counters:

CRCX	:4	MDCX	:9	DLCX	:2
AUEP	:1	AUCX	:0	NTFY	:43
RSIP	:79	EPCF	:0	RQNT	:51
000-199	:0	200-299	:95	300-999	:0

Meaning

The output displays information about all MGCP counters. Verify the following information:

- Summary of MGCP counters
- MGCP error counters
- MGCP packet counters

SEE ALSO

| [Understanding the MGCP ALG | 247](#)

Example: Configuring Three-Zone ISP-Hosted Service Using MGCP ALG and NAT

IN THIS SECTION

- [Requirements | 265](#)
- [Overview | 266](#)
- [Configuration | 267](#)
- [Verification | 279](#)

This example shows how to configure a three-zone configuration using MGCP ALG and NAT.

Requirements

Before you begin, understand NAT support with MGCP ALG. See "[Understanding the MGCP ALG](#)" on [page 247](#).

Overview

IN THIS SECTION

- [Topology](#) | [266](#)

Typically, a three-zone configuration is used when an ISP in one geographical location provides service to two networks in different geographical locations.

In this example (see [Figure 24 on page 267](#)), an ISP located on the USA West Coast provides MGCP service to customers in separate networks in Asia and San Francisco. Asia customers are in the asia-3 zone and are supported by the asia-gw gateway; San Francisco customers are in the sf-2 zone and are supported by the sf-gw gateway. A call agent, west-ca, is in the DMZ. The gateways and the call agent are listed in [Table 14 on page 267](#), showing the corresponding IP address, interface, and zone.

In this example, after creating zones and setting addresses for the gateways and the call agent, you associate the zones to interfaces, and then configure static NAT to the call agent and source NAT for communication from an IP phone in the sf-2 zone to phones in the asia-3 zone. You also configure a policy between the zones to allow the communication.

Topology

[Figure 24 on page 267](#) shows a three-zone ISP-hosted service.

Figure 24: Three-Zone ISP-Hosted Service

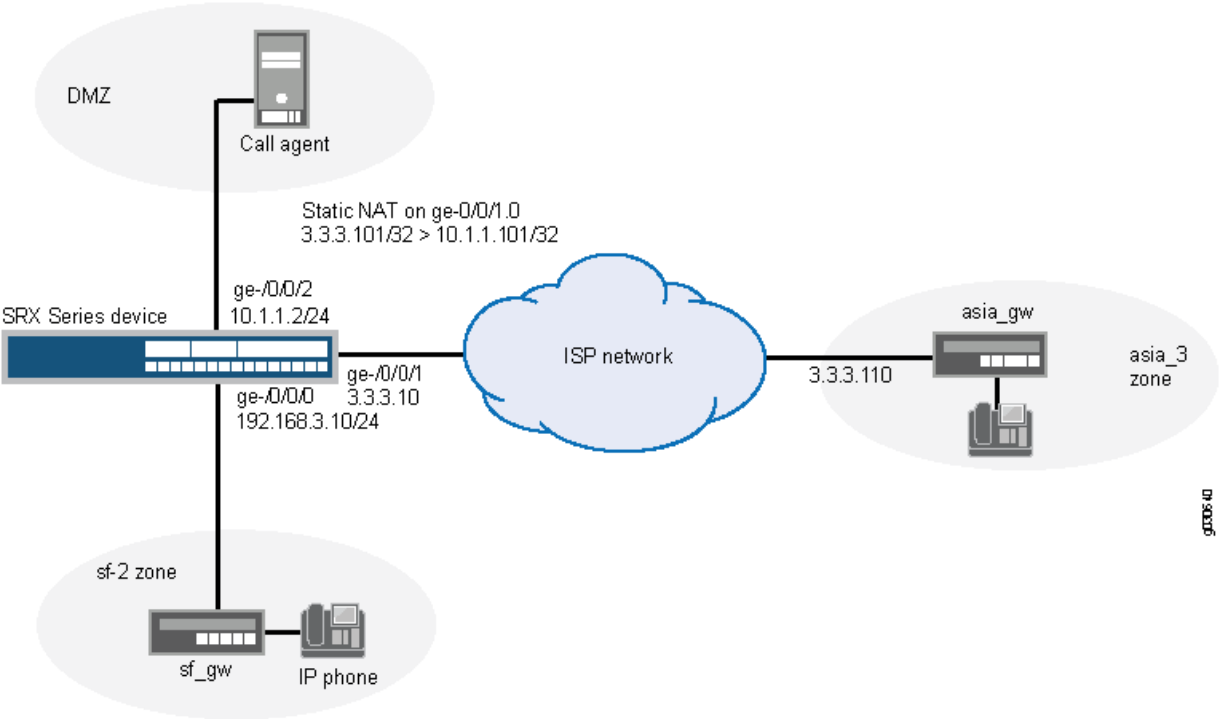


Table 14: Three-Zone ISP-Host Service

Gateway	IP Address	Interface	Zone
sf-gw	192.168.3.201	ge-0/0/0	sf-2
asia-gw	3.3.3.101	ge-0/0/1	asia-3
west-ca	10.1.1.101	ge-0/0/2	DMZ

Configuration

IN THIS SECTION

- Procedure | 268

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 192.168.3.10/24
>set interfaces ge-0/0/1 unit 0 family inet address 3.3.3.10/24
>set interfaces ge-0/0/2 unit 0 family inet address 10.1.1.2/24
>set security zones security-zone sf-2 interfaces ge-0/0/0.0
>set security zones security-zone asia-3 interfaces ge-0/0/1.0
>set security zones security-zone dmz interfaces ge-0/0/2.0
>set security address-book book1 address sf-gw 192.168.3.201/32
>set security address-book book1 attach zone sf-2
>set security address-book book2 address asia-gw 3.3.3.101/32
>set security address-book book2 attach zone asia-3
>set security address-book book3 address west-ca 10.1.1.101/32
>set security address-book book3 attach zone dmz
>set security nat source pool ip-phone-pool address 3.3.3.20/32
>set security nat source rule-set phones from zone sf-2
>set security nat source rule-set phones to zone asia-3
>set security nat source rule-set phones rule phone1 match source-address 192.168.3.10/32
>set security nat source rule-set phones rule phone1 match destination 3.3.3.101/32
>set security nat source rule-set phones rule phone1 then source-nat pool ip-phone-pool
>set security nat static rule-set to-callagent from zone asia-3
>set security nat static rule-set to-callagent rule phone1 match destination-address
3.3.3.101/32
>set security nat static rule-set to-callagent rule phone1 then static-nat prefix 10.1.1.101/32
>set security nat proxy-arp interface ge-0/0/1.0 address 3.3.3.101/32
>set security nat proxy-arp interface ge-0/0/1.0 address 3.3.3.20/32
>set security policies from-zone dmz to-zone asia-3 policy pol-dmz-to-asia-3 match source-
address west-ca
>set security policies from-zone dmz to-zone asia-3 policy pol-dmz-to-asia-3 match destination-
address asia-gw
>set security policies from-zone dmz to-zone asia-3 policy pol-dmz-to-asia-3 match application
junos-mgcp
>set security policies from-zone dmz to-zone asia-3 policy pol-dmz-to-asia-3 then permit
>set security policies from-zone asia-3 to-zone dmz policy pol-asia-3-to-dmz match source-
address asia-gw
```

```

>set security policies from-zone asia-3 to-zone dmz policy pol-asia-3-to-dmz match destination-
address west-ca
>set security policies from-zone asia-3 to-zone dmz policy pol-asia-3-to-dmz match application
junos-mgcp
>set security policies from-zone asia-3 to-zone dmz policy pol-asia-3-to-dmz then permit
>set security policies from-zone sf-2 to-zone dmz policy pol-sf-2-to-dmz match source-address sf-
gw
>set security policies from-zone sf-2 to-zone dmz policy pol-sf-2-to-dmz match destination-
address west-ca
>set security policies from-zone sf-2 to-zone dmz policy pol-sf-2-to-dmz match application junos-
mgcp
>set security policies from-zone sf-2 to-zone dmz policy pol-sf-2-to-dmz then permit
>set security policies from-zone dmz to-zone sf-2 policy pol-dmz-to-sf-2 match source-address
west-ca
>set security policies from-zone dmz to-zone sf-2 policy pol-dmz-to-sf-2 match destination-
address sf-gw
>set security policies from-zone dmz to-zone sf-2 policy pol-dmz-to-sf-2 match application junos-
mgcp
>set security policies from-zone dmz to-zone sf-2 policy pol-dmz-to-sf-2 then permit
>set security policies from-zone sf-2 to-zone asia-3 policy pol-sf-2-to-asia-3 match source-
address sf-gw
>set security policies from-zone sf-2 to-zone asia-3 policy pol-sf-2-to-asia-3 match destination-
address asia-gw
>set security policies from-zone sf-2 to-zone asia-3 policy pol-sf-2-to-asia-3 match application
junos-mgcp
>set security policies from-zone sf-2 to-zone asia-3 policy pol-sf-2-to-asia-3 then permit
>set security policies from-zone asia-3 to-zone sf-2 policy pol-asia-3-to-sf-2 match source-
address asia-gw
>set security policies from-zone asia-3 to-zone sf-2 policy pol-asia-3-to-sf-2 match destination
sf-gw
>set security policies from-zone asia-3 to-zone sf-2 policy pol-asia-3-to-sf-2 match application
junos-mgcp
>set security policies from-zone asia-3 to-zone sf-2 policy pol-asia-3-to-sf-2 then permit
>set security policies from-zone sf-2 to-zone sf-2 policy pol-intra-sf-2 match source-address
any
>set security policies from-zone sf-2 to-zone sf-2 policy pol-intra-sf-2 match destination-
address any
>set security policies from-zone sf-2 to-zone sf-2 policy pol-intra-sf-2 match application any
> set security policies from-zone sf-2 to-zone sf-2 policy pol-intra-sf-2 then permit
>set security policies from-zone asia-3 to-zone asia-3 policy pol-intra-asia-3 match source-
address any
>set security policies from-zone asia-3 to-zone asia-3 policy pol-intra-asia-3 match destination-
address any

```

```
>set security policies from-zone asia-3 to-zone asia-3 policy pol-intra-asia-3 match application
any
>set security policies from-zone asia-3 to-zone asia-3 policy pol-intra-asia-3 then permit
```

Step-by-Step Procedure

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

To configure a three-zone configuration using MGCP ALG and NAT:

1. Configure interfaces.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 192.168.3.10/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 3.3.3.10/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 10.1.1.2/24
```

2. Create security zones.

```
[edit security zones]
user@host# set security-zone sf-2 interfaces ge-0/0/0
user@host# set security-zone asia-3 interfaces ge-0/0/1
user@host# set security-zone dmz interfaces ge-0/0/2
```

3. Create address books and assign zones to them.

```
[edit security address-book book1]
user@host# set address sf-gw 192.168.3.201/32
user@host# set attach zone sf-2
```

```
[edit security address-book book2]
user@host# set address asia-gw 3.3.3.101/32
user@host# set attach zone asia-3
```

```
[edit security address-book book3]
user@host# set address west-ca 10.1.1.101/32
user@host# set attach zone dmz
```

4. Create a static NAT rule set and set the match conditions and actions for it.

```
[edit security nat static rule-set to-callagent]
user@host# set from zone asia-3
user@host# set rule phone1 match destination-address 3.3.3.101/32
user@host# set rule phone1 then static-nat prefix 10.1.1.101/32
```

5. Configure proxy ARP for address 3.3.3.101/32 on interface ge-0/0/1.0.

```
[edit security nat ]
user@host# set proxy-arp interface ge-0/0/1.0 address 3.3.3.101/32
```

6. Create a source NAT pool.

```
[edit security nat]
user@host# set source pool ip-phone-pool address 3.3.3.20/32
```

7. Create a source NAT rule set and set the match conditions and actions for it.

```
[edit security nat source rule-set phones]
user@host# set from zone sf-2
user@host# set to zone asia-3
user@host# set rule phone1 match source-address 192.168.3.10/32
user@host# set rule phone1 match destination-address 3.3.3.101/32
user@host# set rule phone1 then source-nat pool ip-phone-pool
```

8. Configure proxy ARP for address 3.3.3.20/32 on interface ge-0/0/1.0.

```
[edit security nat ]
user@host# set proxy-arp interface ge-0/0/1.0 address 3.3.3.20/32
```

9. Configure a policy to allow traffic from DMZ to Asia.

```
[edit security policies from-zone dmz to-zone asia-3 policy pol-dmz-to-asia-3]
user@host# set match source-address west-ca
user@host# set match destination-address asia-gw
user@host# set match application junos-mgcp
user@host# set then permit
```

10. Configure a policy to allow traffic from Asia to DMZ.

```
[edit security policies from-zone asia-3 to-zone dmz policy pol-asia-3-to-dmz]
user@host# set match source-address asia-gw
user@host# set match destination-address west-ca
user@host# set match application junos-mgcp
user@host# set then permit
```

11. Configure a policy to allow traffic from San Francisco to DMZ.

```
[edit security policies from-zone sf-2 to-zone dmz policy pol-sf-2-to-dmz]
user@host# set match source-address sf-gw
user@host# set match destination-address west-ca
user@host# set match application junos-mgcp
user@host# set then permit
```

12. Configure a policy to allow traffic from DMZ to San Francisco.

```
[edit security policies from-zone dmz to-zone sf-2 policy pol-dmz-to-sf-2]
user@host# set match source-address west-ca
user@host# set match destination-address sf-gw
user@host# set match application junos-mgcp
user@host# set then permit
```

13. Configure a policy to allow traffic from San Francisco to Asia.

```
[edit security policies from-zone sf-2 to-zone asia-3 policy pol-sf-2-to-asia-3]
user@host# set match source-address sf-gw
user@host# set match destination-address asia-gw
user@host# set match application junos-mgcp
user@host# set then permit
```

14. Configure a policy to allow traffic from Asia to San Francisco.

```
[edit security policies from-zone asia-3 to-zone sf-2 policy pol-asia-3-to-sf-2]
user@host# set match source-address asia-gw
user@host# set match destination-address sf-gw
user@host# set match application junos-mgcp
user@host# set then permit
```

15. Configure a policy to allow traffic on devices within San Francisco.

```
[edit security policies from-zone sf-2 to-zone sf-2 policy pol-intra-sf-2]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application any
user@host# set then permit
```

16. Configure a policy to allow traffic on devices within Asia.

```
[edit security policies from-zone asia-3 to-zone asia-3 policy pol-intra-asia-3]
user@host# set match source-address any
user@host# set match destination-address any
```

```
user@host# set match application any
user@host# set then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, `show security address-book`, `show security nat`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 192.168.3.10/24;
      }
    }
  }
  ge-0/0/1 {
    unit 0 {
      family inet {
        address 3.3.3.10/24;
      }
    }
  }
  ge-0/0/2 {
    unit 0 {
      family inet {
        address 10.1.1.2/24;
      }
    }
  }
[edit]
user@host# show security zones
security-zone sf-2 {
  interfaces {
    ge-0/0/0.0;
  }
}
```

```

security-zone asia-3 {
    interfaces {
        ge-0/0/1.0;
    }
}
security-zone dmz {
    interfaces {
        ge-0/0/2.0;
    }
}
[edit]
user@host# show security address-book
book1 {
    address sf-gw 192.168.3.201/32;
    attach {
        zone sf-2;
    }
}
book2 {
    address asia-gw 3.3.3.101/32;
    attach {
        zone asia-3;
    }
}
book3 {
    address west-ca 10.1.1.101/32;
    attach {
        zone dmz;
    }
}
[edit]
user@host# show security nat
source {

    pool ip-phone-pool {
        address {
            3.3.3.20/32;
        }
    }

    rule-set phones {
        from zone sf-2;
        to zone asia-3;

```



```

    rule phone1 {
        match {
            source-address 192.168.3.10/32;
            destination-address 3.3.3.101/32;
        }
        then {
            source-nat {
                pool {
                    ip-phone-pool;
                }
            }
        }
    }
}

static {
    rule-set to-callagent {
        from zone asia-3;
        rule phone1 {
            match {
                destination-address 3.3.3.101/32;
            }
            then {
                static-nat prefix 10.1.1.101/32;
            }
        }
    }
}

proxy-arp {

    interface ge-0/0/1.0 {
        address {
            3.3.3.101/32;
            3.3.3.20/32;
        }
    }
}

[edit]
user@host# show security policies
from-zone dmz to-zone asia-3 {
    policy pol-dmz-to-asia-3 {
        match {
            source-address west-ca;

```

```

        destination-address asia-gw;
        application junos-mgcp;
    }
    then {
        permit;
    }
}
}
from-zone asia-3 to-zone dmz {
    policy pol-asia-3-to-dmz {
        match {
            source-address asia-gw;
            destination-address west-ca;
            application junos-mgcp;
        }
        then {
            permit;
        }
    }
}
from-zone sf-2 to-zone dmz {
    policy pol-sf-2-to-dmz {
        match {
            source-address sf-gw;
            destination-address west-ca;
            application junos-mgcp;
        }
        then {
            permit;
        }
    }
}
from-zone dmz to-zone sf-2 {
    policy pol-dmz-to-sf-2 {
        match {
            source-address west-ca;
            destination-address sf-gw;
            application junos-mgcp;
        }
        then {
            permit;
        }
    }
}

```

```

}
from-zone sf-2 to-zone asia-3 {
    policy pol-sf-2-to-asia-3 {
        match {
            source-address sf-gw;
            destination-address asia-gw;
            application junos-mgcp;
        }
        then {
            permit;
        }
    }
}
from-zone asia-3 to-zone sf-2 {
    policy pol-asia-3-to-sf-2 {
        match {
            source-address asia-gw;
            destination-address sf-gw;
            application junos-mgcp;
        }
        then {
            permit;
        }
    }
}
from-zone sf-2 to-zone sf-2 {
    policy pol-intra-sf-2 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}
from-zone asia-3 to-zone asia-3 {
    policy pol-intra-asia-3 {
        match {
            source-address any;
            destination-address any;
            application any;

```

```

    }
    then {
        permit;
    }
}
}

```

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

Verification

IN THIS SECTION

- [Verifying MGCP ALG | 279](#)
- [Verifying MGCP Calls | 280](#)
- [Verifying MGCP ALG Statistics | 281](#)
- [Verifying MGCP Endpoints | 282](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying MGCP ALG

Purpose

Verify if the MGCP ALG is enabled.

Action

From operational mode, enter the `show security alg status | match mgcp` command.

```
user@host> show security alg status | match mgcp
```

```
MGCP      : Enabled
```

Meaning

The output shows the MGCPALG status as follows:

- Enabled—Shows the MGCP ALG is enabled.
- Disabled—Shows the MGCP ALG is disabled.

Verifying MGCP Calls

Purpose

Verify the MGCP calls that are currently active.

Action

From operational mode, enter the `show security alg mgcp calls` command.

```
user@host> show security alg mgcp calls
```

Endpoint@GW	Zone	Call ID	RM Group
d001@101.50.10.1	Trust	10d55b81140e0f76	512
Connection Id> 0			
Local SDP> o: 101.50.10.1		x_o: 101.50.10.1	
c: 101.50.10.1/32206		x_c: 101.50.10.1/32206	
Remote SDP> c: 3.3.3.5/16928		x_c: 3.3.3.5/16928	
Endpoint@GW	Zone	Call ID	RM Group
d001@3.3.3.5	Untrust	3a104e9b41a7c4c9	511
Connection Id> 0			
Local SDP> o: 3.3.3.5		x_o: 3.3.3.5	
c: 3.3.3.5/16928		x_c: 3.3.3.5/16928	
Remote SDP> c: 101.50.10.1/32206		x_c: 101.50.10.1/32206	

Meaning

The output displays information about all MGCP calls. Verify the following information:

- Endpoint
- Zone

- Call identifier
- Resource Manager group

Verifying MGCP ALG Statistics

Purpose

Verify the MGCP ALG statistics.

Action

From operational mode, enter the `show security alg mgcp counters` command.

```
user@host> show security alg mgcp counters
```

```
MGCP counters summary:
Packets received           :284
Packets dropped            :0
Message received           :284
Number of connections      :4
Number of active connections :3
Number of calls            :4
Number of active calls     :3
Number of transactions     :121
Number of active transactions:52
Number of re-transmission  :68
MGCP Error Counters:
Unknown-method             :0
Decoding error             :0
Transaction error          :0
Call error                 :0
Connection error           :0
Connection flood drop      :0
Message flood drop         :0
IP resolve error           :0
NAT error                  :0
Resource manager error     :0
MGCP Packet Counters:
CRCX      :4      MDCX      :9      DLCX      :2
AUEP      :1      AUCX      :0      NTFY      :43
```

```

RSIP      :79      EPCF      :0      RQNT      :51
000-199   :0      200-299   :95    300-999   :0

```

Meaning

The output displays information about all MGCP counters. Verify the following information:

- Summary of MGCP counters
- MGCP error counters
- MGCP packet counters

Verifying MGCP Endpoints

Purpose

Verify the MGCP endpoints.

Action

From operational mode, enter the `show security alg mgcp endpoints` command.

```
user@host> show security alg mgcp endpoints
```

```

Gateway: 101.50.10.1 Zone: Trust IP: 101.50.10.1 -> 101.50.10.1
  Endpoint      Trans #  Call #  Notified Entity
  d001          1        1        0.0.0.0/0->0.0.0.0/0
Gateway: 3.3.3.5 Zone: Untrust IP: 3.3.3.5 -> 3.3.3.5
  Endpoint      Trans #  Call #  Notified Entity
  d001          1        1        0.0.0.0/0->0.0.0.0/0

```

Meaning

The output displays information about all MGCP endpoints. Verify the following information:

- Gateway IP address and zone of both endpoints
- Endpoint identifier, transaction number, call number, and notified entity for each gateway

SEE ALSO

Static NAT Configuration Overview

Understanding Source NAT

Understanding MGCP ALG Call Duration and Timeouts

The call duration feature gives you control over Media Gateway Control Protocol (MGCP) call activity and helps you to manage network resources.

Typically a Delete Connection (DLCX) message will be sent out to delete a connection. The MGCP Application Layer Gateway (ALG) intercepts it and removes all media sessions for that connection.

A call can have one or more voice channels. Each voice channel has two sessions (or two media streams), one for Real-Time Transport Protocol (RTP) traffic and one for Real-Time Control Protocol (RTCP) signaling. When managing the sessions, the device considers the sessions in each voice channel as one group. Timeouts and call duration settings apply to a group as opposed to each session.

The following parameters govern MGCP call activity:

- **maximum-call-duration**—This parameter sets the absolute maximum length of a call. When a call exceeds this parameter setting, the MGCP ALG tears down the call and releases the media sessions. The default setting is 720 minutes, and the range is 3 through 720 minutes. This setting also frees up bandwidth in cases where calls fail to properly terminate.
- **inactive-media-timeout**—This parameter indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the MGCP ALG gates opened for media are closed. The default setting is 120 seconds, and the range is 10 through 2550 seconds. Note that upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated.



NOTE: The `inactive-media-timeout` value should be less than the `maximum-call-duration` value.

- **transaction-timeout**—A transaction is a command and its mandatory response. For example, an NOTIFY from the gateway to the call agent or a 200 OK from the call agent to the gateway. The Juniper Networks device tracks these transactions and clears them when they time out. The timeout range for MGCP transactions is 3 through 50 seconds and the default is 30 seconds.

Example: Setting MGCP ALG Call Duration

IN THIS SECTION

- Requirements | 284
- Overview | 284
- Configuration | 284
- Verification | 285

This example shows how to set call duration for the MGCP ALG.

Requirements

Before you begin, determine the type of parameter used to control the MGCP call activity and manage its network resources. See ["Understanding MGCP ALG Call Duration and Timeouts" on page 283](#).

Overview

The `maximum-call-duration` parameter governs MGCP call activity and sets the absolute maximum length of a call. When a call exceeds this parameter setting, the MGCP ALG tears down the call and releases the media sessions. The default setting is 720 minutes, and the range is 3 through 720 minutes. This setting also frees up bandwidth in cases where calls fail to properly terminate. In this example, the call duration is set to 600 minutes.

Configuration

IN THIS SECTION

- Procedure | 285

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To set call duration for the MGCP ALG:

1. Select **Configure > Security > ALG**.
2. Select the **MGCP** tab.
3. In the **Maximum call duration** box, enter **600**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options > Commit**.

Step-by-Step Procedure

To set call duration for the MGCP ALG:

1. Configure the MGCP ALG call duration.

```
[edit]  
user@host# set security alg mgcp maximum-call-duration 600
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg mgcp` command.

Example: Setting MGCP ALG Inactive Media Timeout

IN THIS SECTION

- [Requirements | 286](#)
- [Overview | 286](#)
- [Configuration | 286](#)
- [Verification | 287](#)

This example shows how to set the inactive media timeout value for the MGCP ALG.

Requirements

Before you begin, determine the type of parameter used to control the MGCP call activity and manage its network resources. See ["Understanding MGCP ALG Call Duration and Timeouts" on page 283](#).

Overview

The `inactive-media-timeout` parameter governs MGCP call activity and indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the MGCP ALG gates opened for media are closed. The default setting is 120 seconds, and the range is from 10 to 2550 seconds. Note that upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated. In this example, the inactive media timeout is set to 90 seconds.

Configuration

IN THIS SECTION

- [Procedure | 287](#)

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To set the inactive media timeout for the MGCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. In the **Inactive Media Timeout** box, enter **90**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To set the inactive media timeout for the MGCP ALG:

1. Configure the MGCP ALG inactive media timeout value.

```
[edit]  
user@host# set security alg mgcp inactive-media-timeout 90
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg mgcp` command.

Example: Setting MGCP ALG Transaction Timeout

IN THIS SECTION

- Requirements | 288
- Overview | 288
- Configuration | 288
- Verification | 289

This example shows how to set the transaction timeout for the MGCP ALG.

Requirements

Before you begin, determine the type of parameter used to control the MGCP call activity and manage its network resources. See ["Understanding MGCP ALG Call Duration and Timeouts" on page 283](#).

Overview

The transaction-timeout parameter governs MGCP call activity and is a signaling message; for example, a NTFY from the gateway to the call agent or a 200 OK from the call agent to the gateway. The Juniper Networks device tracks these transactions, and clears them when they time out. The timeout range for MGCP transactions is from 3 to 50 seconds, and the default is 30 seconds. In this example, the transaction timeout is set to 20 seconds.

Configuration

IN THIS SECTION

- Procedure | 289

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To set the transaction timeout for the MGCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. In the **Transaction Timeout** box, enter **20**.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To set the transaction timeout for the MGCP ALG:

1. Configure the MGCP ALG transaction timeout value.

```
[edit]  
user@host# set security alg mgcp transaction-timeout 20
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg mgcp` command.

Example: Configuring MGCP ALG DoS Attack Protection

IN THIS SECTION

- [Requirements | 290](#)
- [Overview | 290](#)
- [Configuration | 290](#)
- [Verification | 291](#)

This example shows how to configure connection flood protection for the MGCP ALG.

Requirements

Before you begin, determine whether to protect the MGCP media gateway from DoS flood attacks.

Overview

You can protect the Media Gateway Control Protocol (MGCP) media gateway from denial-of-service (DoS) flood attacks by limiting the number of remote access service (RAS) messages and connections per second it will attempt to process.

When you configure MGCP message flood protection, the MGCP Application Layer Gateway (ALG) drops any messages exceeding the threshold you set. The range is 2 to 50,000 messages per second per media gateway, and the default is 1000 messages per second per media gateway.

When you configure MGCP connection flood protection, the MGCP ALG drops any connection request exceeding the threshold you set. This limits the rate of processing of CreateConnection (CRCX) commands, thereby indirectly limiting pinhole creation. The range is 2 to 10,000 connection requests per second per media gateway, the default is 200.

In this example, you configure the MGCP ALG to drop any message requests exceeding 10,000 requests per second and to drop any connection requests exceeding 4000 per second.

Configuration

IN THIS SECTION

- [Procedure | 291](#)

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To configure connection flood protection for the MGCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. In the **Message flood gatekeeper threshold** box, type **10000**.
4. In the **Connection flood threshold** box, type **4000**.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure connection flood protection for the MGCP ALG:

1. Configure the connection flood threshold value.

```
[edit]
user@host# set security alg mgcp application-screen message-flood threshold 10000
user@host# set security alg mgcp application-screen connection-flood threshold 4000
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg mgcp` command.

Example: Allowing Unknown MGCP ALG Message Types

IN THIS SECTION

- [Requirements | 292](#)
- [Overview | 292](#)
- [Configuration | 293](#)
- [Verification | 294](#)

This example shows how to configure the MGCP ALG to allow unknown MGCP message types in both NAT mode and route mode.

Requirements

Before you begin, determine whether to accommodate new and unknown MGCP message types for the device.

Overview

To accommodate on-going development of the Media Gateway Control Protocol (MGCP), you might want to allow traffic containing new MGCP message types. The unknown MGCP message type feature enables you to configure the Juniper Networks device to accept MGCP traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.

This feature enables you to specify how unidentified MGCP messages are handled by the Juniper Networks device. The default is to drop unknown (unsupported) messages. Unknown messages can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment. Permitting unknown MGCP messages can help you get your network operational so that you can later analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped.

Note that this command applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

Configuration

IN THIS SECTION

- [Procedure](#) | 293

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To configure the MGCP ALG to allow unknown message types:

1. Select **Configure>Security>ALG**.
2. Select the **MGCP** tab.
3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure the MGCP ALG to allow unknown message types:

1. Allow unknown message types to pass if the session is in either NAT mode or in route mode.

[edit]

```
user@host# set security alg mgcp application-screen unknown-message permit-nat-applied permit-routed
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg mgcp` command.

RELATED DOCUMENTATION

[Understanding VoIP ALG Types | 188](#)

[VoIP DSCP Rewrite Rules | 189](#)

[SIP ALG | 326](#)

SCCP ALG

IN THIS SECTION

- [Understanding SCCP ALGs | 295](#)
- [SCCP ALG Configuration Overview | 302](#)
- [Example: Setting SCCP ALG Inactive Media Timeouts | 302](#)
- [Example: Allowing Unknown SCCP ALG Message Types | 304](#)
- [Example: Configuring SCCP ALG DoS Attack Protection | 306](#)
- [Example: Configuring the SCCP ALG Call Manager or TFTP Server in the Private Zone | 308](#)
- [Verifying SCCP ALG Configurations | 321](#)

The Skinny Client Control Protocol (SCCP) is a simple and lightweight protocol requiring relatively little computer processing. SCCP clients use TCP/IP to communicate with Call Manager applications in a cluster.

Understanding SCCP ALGs

IN THIS SECTION

- [SCCP Security | 296](#)
- [SCCP Components | 296](#)
- [SCCP Transactions | 297](#)
- [SCCP Version | 298](#)
- [SCCP Control Messages and RTP Flow | 299](#)
- [SCCP Messages | 300](#)

Skinny Client Control Protocol (SCCP) is a Cisco proprietary protocol for call signaling. Skinny is based on a call-agent-based call-control architecture. The control protocol uses binary-coded frames encoded on TCP frames sent to well-known TCP port number destinations to set up and tear down RTP media sessions.

The SCCP protocol, in the same way as other call control protocols, negotiates media endpoint parameters—specifically the Real-Time Transport Protocol (RTP) port number and the IP address of media termination—by embedding information in the control packets. The SCCP Application Layer Gateway (ALG) parses these control packets and facilitates media and control packets to flow through the system.

The SCCP ALG also implements rate limiting of calls and helps protect critical resources from overloading and denial-of-service (DoS) attacks.

The following functions are implemented by the SCCP ALG in Junos OS:

- Validation of SCCP protocol data units
- Translation of embedded IP address and port numbers
- Allocation of firewall resources (pinholes and gates) to pass media
- Aging out idle calls
- Configuration API for SCCP ALG parameters
- Operational mode API for displaying counters, status and statistics

In the SCCP architecture, a proxy, known as the Call Manager, does most of the processing. IP phones, also called End Stations, run the SCCP client and connect to a primary (and, if available, a secondary) Call

Manager over TCP on port 2000 and register with the primary Call Manager. This connection is then used to establish calls coming to or from the client.

The SCCP ALG supports the following:

- Call flow from a SCCP client, through the Call Manager, to another SCCP client.
- Seamless failover—Switches over all calls in process to the standby firewall during failure of the primary.
- Voice-over-IP (VoIP) signaling payload inspection—Fully inspects the payload of incoming VoIP signaling packets. Any malformed packet attack is blocked by the ALG.
- SCCP signaling payload inspection—Fully inspects the payload of incoming SCCP signaling packets. Any malformed-packet attack is blocked by the ALG.
- Stateful processing—Invokes the corresponding VoIP-based state machines to process the parsed information. Any out-of-state or out-of-transaction packet is identified and properly handled.
- Network Address Translation (NAT)—Translates any embedded IP address and port information in the payload, based on the existing routing information and network topology, with the translated IP address and port number, if necessary.
- Pinhole creation and management for VoIP traffic—Identifies IP address and port information used for media or signaling and dynamically opens (and closes) pinholes to securely stream the media.

This topic includes the following sections:

SCCP Security

The SCCP ALG includes the following security features:

- Stateful inspection of SCCP control messages over TCP and validation of the message format, and message validity for the current call state. Invalid messages are dropped.
- Security policy enforcement between Cisco IP phones and Cisco Call Manager.
- Protect against call flooding by rate limiting the number of calls processed by the ALG.
- Seamless failover of calls, including the ones in progress in case of device failure in a clustered deployment.

SCCP Components

The principal components of the SCCP VoIP architecture include the following:

SCCP Client

The SCCP client runs on an IP phone, also called an *End Station*, which uses SCCP for signaling and for making calls. For an SCCP client to make a call, it must first register with a Primary Call Manager (and a secondary, if available). The connection between the client and the Call Manager is over TCP on port 2000. This connection is then used to establish calls to or from the client. Transmission of media is over RTP, UDP, and IP.

Call Manager

The Call Manager implements SCCP call control server software and has overall control of all devices and communication in the SCCP VoIP network. Its functions include defining, monitoring and controlling SCCP groups, regions of numbers, and route plans; providing initialization, admission, and registration of devices on the network; providing a redundant database that contains addresses, phone numbers, and number formats; and initiating contact with called devices or their agents to establish logical sessions in which voice communication can flow.

Cluster

A *cluster* is a collection of SCCP clients and a Call Manager. The Call Manager in the cluster detects all SCCP clients in the cluster. There can be more than one Call Manager for backup in a cluster. Call Manager behavior varies in each of the following cluster scenarios:

- Intra-Cluster, in which the Call Manager detects each SCCP client, and the call is between SCCP clients of the same cluster.
- Inter-Cluster, in which the Call Manager needs to communicate with another Call Manager using H.323 for call setup.
- Inter-Cluster calls using the gatekeeper for admission control and address resolution.

Call Manager behavior also varies with calls between an SCCP client and a phone in a public switched telephone network (PSTN), and with calls between an SCCP client and a phone in another administrative domain that is using H.323.

SCCP Transactions

SCCP transactions are the processes that need to take place in order for an SCCP call to proceed. SCCP transactions include the following processes:

Client Initialization

To initialize, the SCCP client needs to determine the IP address of the Call Manager, its own IP address, and other information about the IP gateway and DNS servers. Initialization takes place on the local LAN. The client sends a Dynamic Host Control Protocol (DHCP) request to get an IP address, the DNS server address, and the TFTP server name and address. The client needs the TFTP server name to download the configuration file called *sep macaddr .cnf*. If the TFTP name is not given, the client uses the default filename in the IP phone. The client then downloads the .cnf (xml) configuration file from TFTP server. CNF files contain the IP address or addresses of the primary and secondary Cisco Call Manager. With this information, the client contacts the Call Manager to register.

Client Registration

The SCCP client, after initialization, registers with the Call Manager over a TCP connection on well-known default port 2000. The client registers by providing the Call Manager with its IP address, the MAC address of the phone, and other information, such as protocol and version. The client cannot initiate or receive calls until it is registered. Keepalive messages keep this TCP connection open between the client and Call Manager so that the client can initiate or receive calls at any time, provided that a policy on the device allows this.

Call Setup

IP phone-to-IP phone call setup using SCCP is always handled by the Call Manager. Messages for call setup are sent to the Call Manager, which returns messages appropriate to the status of the call. If call setup is successful, and a policy on the device allows the call, the Call Manager sends the media setup messages to the client.

Media Setup

The Call Manager sends the IP address and port number of the called party to the calling party. The Call Manager also sends the media IP address and port number of the calling party to the called party. After media setup, media is transmitted directly between clients. When the call ends, the Call Manager is informed and terminates the media streams. At no time during this process does the Call Manager hand over call-setup function to the client. Media is streamed directly between clients through RTP/UDP/IP.

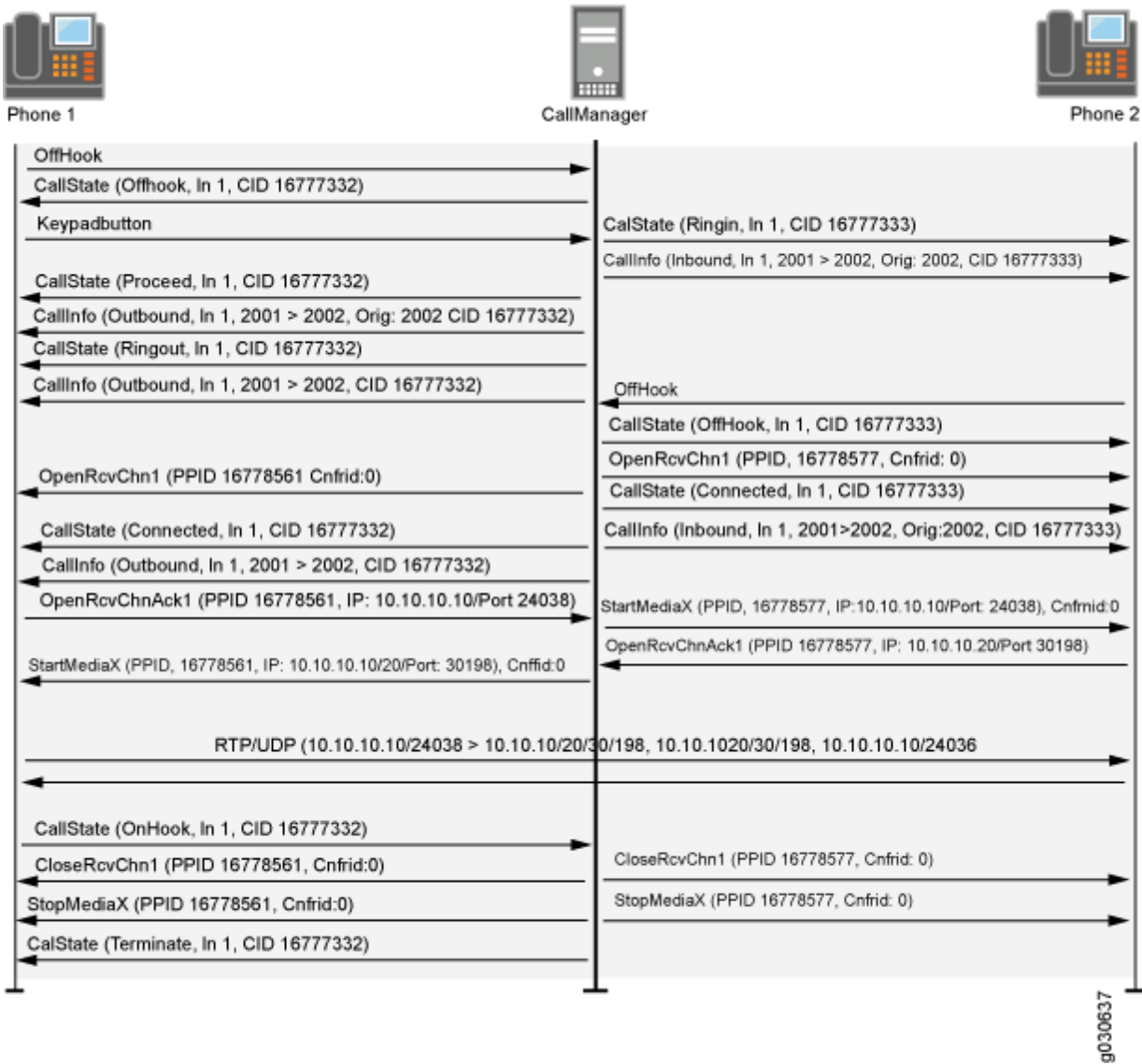
SCCP Version

Starting in Junos OS Release 12.1X46-D10 and Junos OS Release 17.3R1, the SCCP ALG supports SCCP versions 16, 17, and 20 and several SCCP messages have been updated with a new format. Cisco Call Manager (CM) version 7 uses SCCP version 20.

SCCP Control Messages and RTP Flow

Figure 25 on page 299 shows the SCCP control messages used to set up and tear down a simple call between Phone 1 and Phone 2. Except for the OffHook message initiating the call from Phone1 and the OnHook message signaling the end of the call, all aspects of the call are controlled by the Call Manager.

Figure 25: Call Setup and Teardown



SCCP Messages

Table 15 on page 300, Table 16 on page 300, Table 17 on page 300, and Table 18 on page 301 list the SCCP call message IDs in the four intervals allowed by the device.

Table 15: Station to Call Manager Messages

#define STATION_REGISTER_MESSAGE	0x00000001
#define STATION_IP_PORT_MESSAGE	0x00000002
#define STATION_ALARM_MESSAGE	0x00000020
#define STATION_OPEN_RECEIVE_CHANNEL_ACK	0x00000022

Table 16: Call Manager to Station Messages

#define STATION_START_MEDIA_TRANSMISSION	0x00000001
#define STATION_STOP_MEDIA_TRANSMISSION	0x00000002
#define STATION_CALL_INFO_MESSAGE	0x00000020
#define STATION_OPEN_RECEIVE_CHANNEL_ACK	0x00000022
#define STATION_CLOSE_RECEIVE_CHANNEL	0x00000106

Table 17: Call Manager 4.0 Messages and Post Sccp 6.2

#define STATION_REGISTER_TOKEN_REQ_MESSAGE	0x00000029
#define STATION_MEDIA_TRANSMISSION_FAILURE	0x0000002A
#define STATION_OPEN_MULTIMEDIA_RECEIVE_CHANNEL_ ACK	0x00000031

Table 18: Call Manager to Station

#define STATION_OPEN_MULTIMEDIA_RECEIVE_CHANNEL	0x00000131
#define STATION_START_MULTIMEDIA_TRANSMISSION	0x00000132
#define STATION_STOP_MULTIMEDIA_TRANSMISSION	0x00000133
#define STATION_CLOSE_MULTIMEDIA_RECEIVE_CHANNEL	0x00000136

SCCP ALG Limitations

- The SCCP is a Cisco proprietary protocol. So, any changes to the protocol by Cisco cause the SCCP ALG implementation to break. However, workarounds are provided to bypass strict decoding and allow any protocol changes to be handled gracefully.
- Any changes to the policies will drop the sessions and impact already established SCCP calls.
- The SCCP ALG opens pinholes that are collapsed during traffic or media inactivity. This means that during a temporary loss of connectivity, media sessions are not reestablished.
- Call Manager (CM) version 6.x and later does not support TCP probe packets in chassis cluster mode. As a result, the existing SCCP sessions will break when there is a failover. You can still create new SCCP sessions during failover.

Understanding SCCP ALG Inactive Media Timeouts

The inactive media timeout feature helps you to conserve network resources and maximize throughput.

This parameter indicates the maximum length of time (in seconds) a call can remain active without any media traffic within a group. Each time a Real-Time Transport Protocol (RTP) or Real-Time Control Protocol (RTCP) packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the gates the Skinny Client Control Protocol (SCCP) opened for media are closed. The default setting is 120 seconds, and the range is from 10 to 2550 seconds. Note that upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated.

SCCP ALG Configuration Overview

The Skinny Client Control Protocol Application Layer Gateway (SCCP ALG) is enabled by default on the device—no action is required to enable it. However, you might choose to fine-tune SCCP ALG operations by using the following instructions:

1. Conserve network resources and maximize throughput. For instructions, see ["Example: Setting SCCP ALG Inactive Media Timeouts" on page 302](#).
2. Enable unknown messages to pass when the session is in Network Address Translation (NAT) mode and route mode. For instructions, see ["Example: Allowing Unknown SCCP ALG Message Types" on page 304](#).
3. Protect the SCCP clients from denial-of-service (DoS) flood attacks. For instructions, see ["Example: Configuring SCCP ALG DoS Attack Protection" on page 306](#).

Example: Setting SCCP ALG Inactive Media Timeouts

IN THIS SECTION

- [Requirements | 302](#)
- [Overview | 302](#)
- [Configuration | 303](#)
- [Verification | 304](#)

This example shows how to set the inactive media timeout value for the SCCP ALG.

Requirements

Before you begin, review the parameter used to indicate the maximum length of time (in seconds) a call can remain active without any media traffic within a group. See ["Understanding SCCP ALG Inactive Media Timeouts" on page 295](#).

Overview

Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the gates the SCCP opened for media are closed. This example sets the media inactivity timeout to 90 seconds.

Configuration

IN THIS SECTION

- [Procedure](#) | 303

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To set the inactive media timeout for the SCCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **SCCP** tab.
3. In the Inactive Media Timeout box, enter 90.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To set the inactive media timeout for the SCCP ALG:

1. Configure the SCCP ALG inactive media timeout value.

```
[edit]  
user@host# set security alg sccp inactive-media-timeout 90
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg sccp` command.

Example: Allowing Unknown SCCP ALG Message Types

IN THIS SECTION

- [Requirements | 304](#)
- [Overview | 304](#)
- [Configuration | 305](#)
- [Verification | 306](#)

This example shows how to configure the SCCP ALG to allow unknown SCCP message types in both NAT mode and route mode.

Requirements

Before you begin, determine whether to accommodate new and unknown SCCP message types for the device.

Overview

To accommodate on-going development of the Skinny Client Control Protocol (SCCP), you might want to allow traffic containing new SCCP message types. The unknown SCCP message type feature enables you to configure the device to accept SCCP traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.

The default is to drop unknown (unsupported) messages. We do not recommend permitting unknown messages because they can compromise security. However, in a secure test or production environment, the unknown SCCP message type command can be useful for resolving interoperability issues with disparate vendor equipment. Permitting unknown SCCP messages can help you get your network operational so that you can later analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped.

Note that the unknown SCCP message type command applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as

a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

Configuration

IN THIS SECTION

- [Procedure](#) | 305

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To configure the SCCP ALG to allow unknown message types:

1. Select **Configure>Security>ALG**.
2. Select the **SCCP** tab.
3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure the SCCP ALG to allow unknown message types:

1. Allow unknown message types to pass if the session is in either NAT mode or in route mode.

```
[edit]
user@host# set security alg sccp application-screen unknown-message permit-nat-applied
permit-routed
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg sccp` command.

Example: Configuring SCCP ALG DoS Attack Protection

IN THIS SECTION

- [Requirements | 306](#)
- [Overview | 306](#)
- [Configuration | 307](#)
- [Verification | 308](#)

This example shows how to configure connection flood protection for the SCCP ALG.

Requirements

Before you begin, determine whether to protect the SCCP media gateway from DoS flood attacks.

Overview

You can protect Skinny Client Control Protocol Application Layer Gateway (SCCP ALG) clients from denial-of-service (DoS) flood attacks by limiting the number of calls they attempt to process.

When you configure SCCP call flood protection, the SCCP ALG drops any calls exceeding the threshold you set. The range is 2 to 1000 calls per second per client, the default is 20.

In this example, the device is configured to drop any calls exceeding 500 per second per client.

Configuration

IN THIS SECTION

- [Procedure](#) | 307

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To configure call flood protection for the SCCP ALG:

1. Select **Configure>Security>ALG**.
2. Select the **SCCP** tab.
3. In the Call flood threshold box, type 500.
4. Click **OK** to check your configuration and save it as a candidate configuration.
5. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure call flood protection for the SCCP ALG:

1. Configure the DoS attack protection:

```
[edit]  
user@host# set security alg sccp application-screen call-flood threshold 500
```

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

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


Verification

To verify the configuration is working properly, enter the `show security alg sccp` command.

Example: Configuring the SCCP ALG Call Manager or TFTP Server in the Private Zone

IN THIS SECTION

- Requirements | 308
- Overview | 308
- Configuration | 311
- Verification | 317

This example shows how to configure static NAT on the outgoing interface of a Juniper Networks device to allow callers in a public zone to register with an SCCP ALG Call Manager or a TFTP server located in a private zone.

Requirements

Before you begin, understand NAT support with SCCP ALG. See ["Understanding SCCP ALGs" on page 295](#).

Overview

IN THIS SECTION

- Topology | 309

In this example (see [Figure 26 on page 310](#)), a single device is serving as a Call Manager or a TFTP server. The Call Manager or TFTP server and phone1 are attached to the private zone, and phone2 is attached to the public zone. You configure a static NAT rule set for the Call Manager or TFTP server so that when phone2 boots up it contacts the TFTP server and obtains the IP address of the Call Manager.

You then create a policy called in-pol to allow SCCP traffic from the public to the private zone and a policy called out-pol to allow phone1 to call out.

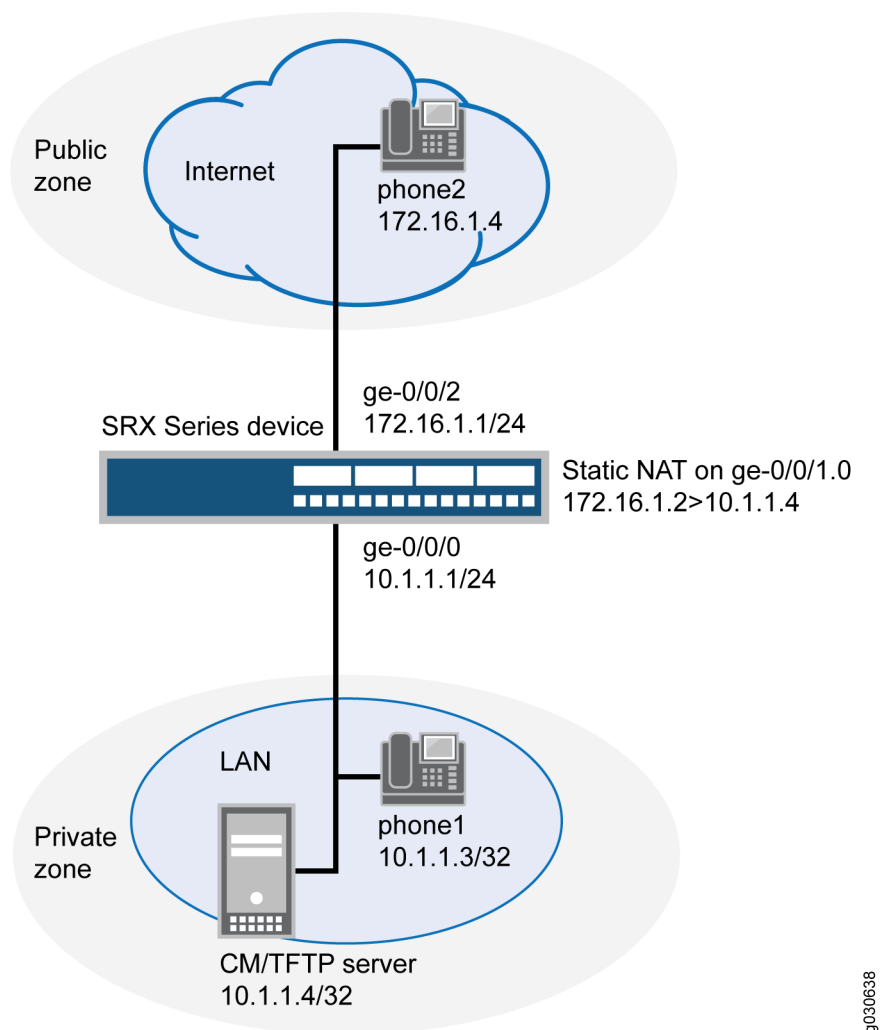


NOTE: We recommend that you change the IP address of the Call Manager, which resides in the TFTP server configuration file (sep <mac_addr>.cnf), to the NAT IP address of the Call Manager.

Topology

[Figure 26 on page 310](#) shows call manager or TFTP server in the private zone.

Figure 26: Call Manager or TFTP Server in the Private Zone



In this example, you configure NAT as follows:

- Create a static NAT rule set called **to-proxy** with a rule called **phone2** to match packets from the public zone with the destination address **172.16.1.2/32**. For matching packets, the destination IP address is translated to the private address **10.1.1.4/32**.
- Configure proxy ARP for the address **172.16.1.2/32** on interface **ge-0/0/1.0**. This allows the system to respond to ARP requests received on the interface for these addresses.
- Configure a second rule set called **phones** with a rule called **phone1** to enable interface NAT for communication from **phone1** to **phone2**.

Configuration

IN THIS SECTION

- [Procedure | 311](#)

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public interfaces ge-0/0/1.0
set security address-book book1 address phone1 10.1.1.3/32
set security address-book book1 address cm-tftp_server 10.1.1.4/32
set security address-book book1 attach zone private
set security address-book book2 address phone2 172.16.1.4/32
set security address-book book2 attach zone public
set security nat source rule-set phones from zone private
set security nat source rule-set phones to zone public
set security nat source rule-set phones rule phone1 match source-address 10.1.1.3/32
set security nat source rule-set phones rule phone1 then source-nat interface
set security nat static rule-set to-proxy from zone public
set security nat static rule-set to-proxy rule phone2 match destination-address 172.16.1.2/32
set security nat static rule-set to-proxy rule phone2 then static-nat prefix 10.1.1.4/32
set security nat proxy-arp interface ge-0/0/1.0 address 172.16.1.2/32
set security policies from-zone public to-zone private policy in-pol match source-address
phone2
set security policies from-zone public to-zone private policy in-pol match destination-address
cm-tftp_server
set security policies from-zone public to-zone private policy in-pol match destination-address
phone1
set security policies from-zone public to-zone private policy in-pol match application junos-
```

```

sccp
set security policies from-zone public to-zone private policy in-pol then permit
set security policies from-zone private to-zone public policy out-pol match source-address any
set security policies from-zone private to-zone public policy out-pol match destination-address
phone2
set security policies from-zone private to-zone public policy out-pol match application junos-
sccp
set security policies from-zone private to-zone public policy out-pol then permit
set security policies from-zone private to-zone private policy tftp-pol match source-address any
set security policies from-zone private to-zone private policy tftp-pol match destination-
address any
set security policies from-zone private to-zone private policy tftp-pol match application junos-
tftp
set security policies from-zone private to-zone private policy tftp-pol then permit

```

Step-by-Step Procedure

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

To configure NAT for an SCCP ALG Call Manager or a TFTP server located in a private zone:

1. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 172.16.1.1/24

```

2. Create security zones.

```

[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/1.0

```

3. Create address books and attach zones to them.

```

[edit security address-book book1]
user@host# set address phone1 10.1.1.3/32

```

```
user@host# set address cm-tftp_server 10.1.1.4/32
user@host# set attach zone private
```

```
[edit security address-book book2]
user@host# set address phone2 172.16.1.4/32
user@host# set attach zone public
```

4. Create a rule set for static NAT and assign a rule to it.

```
[edit security nat static rule-set to-proxy]
user@host# set from zone public
user@host# set rule phone2 match destination-address 172.16.1.2/32
user@host# set rule phone2 then static-nat prefix 10.1.1.4/32
```

5. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/1.0 address 172.16.1.2/32
```

6. Configure interface NAT for communication from phone1 to phone2.

```
[edit security nat source rule-set phones]
user@host# set from zone private
user@host# set to zone public
user@host# set rule phone1 match source-address 10.1.1.3/32
user@host# set rule phone1 then source-nat interface
```

7. Configure a policy to allow traffic from the public zone to the private zone.

```
[edit security policies from-zone public to-zone private policy in-pol]
user@host# set match source-address phone2
user@host# set match destination-address cm-tftp_server
user@host# set match destination-address phone1
user@host# set match application junos-sccp
user@host# set then permit
```

8. Configure a policy to allow traffic from the private zone to the public zone.

```
[edit security policies from-zone private to-zone public policy out-pol]
user@host# set match source-address any
user@host# set match destination-address phone2
user@host# set match application junos-sccp
user@host# set then permit
```

9. Configure a policy to allow traffic from phone1 to the CM/TFTP server.

```
[edit security policies from-zone private to-zone private policy tftp-pol]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application junos-tftp
user@host# set then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, `show security address-book`, `show security nat`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 10.1.1.1/24;
      }
    }
  }
  ge-0/0/1 {
    unit 0 {
      family inet {
        address 172.16.1.1/24;
      }
    }
  }
```

```

    }
[edit]
user@host# show security zones
security-zone private {
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone public {
    interfaces {
        ge-0/0/1.0;
    }
}
[edit]
user@host# show security address-book
book1 {
    address phone1 10.1.1.3/32;
    address cm-tftp_server 10.1.1.4/32;
    attach {
        zone private;
    }
}
book2 {
    address phone2 172.16.1.4/32;
    attach {
        zone public;
    }
}
[edit]
user@host# show security nat
source {

    rule-set phones {
        from zone private;
        to zone public;
        rule phone1 {
            match {
                source-address 10.1.1.3/32;
            }
            then {
                source-nat {
                    interface;
                }
            }
        }
    }
}

```



```

    }
  }
}
static {
  rule-set to-proxy {
    from zone public;
    rule phone2 {
      match {
        destination-address 172.16.1.2/32;
      }
      then {
        static-nat prefix 10.1.1.4/32;
      }
    }
  }
}
proxy-arp {
  interface ge-0/0/1.0 {
    address {
      172.16.1.2/32;
    }
  }
}
[edit]
user@host# show security policies
from-zone public to-zone private {
  policy in-pol {
    match {
      source-address phone2;
      destination-address cm-tftp_server;
      destination-address phone1;
      application junos-sccp;
    }
    then {
      permit;
    }
  }
}
from-zone private to-zone public {
  policy out-pol {
    match {

```

```

        source-address any;
        destination-address phone2;
        application junos-sccp;
    }
    then {
        permit;
    }
}
}
from-zone private to-zone private {
    policy tftp-pol {
        match {
            source-address any;
            destination-address any;
            application junos-tftp;
        }
        then {
            permit;
        }
    }
}
}

```

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

Verification

IN THIS SECTION

- [Verifying Source NAT Rule Usage | 318](#)
- [Verifying Static NAT Configuration | 318](#)
- [Verifying SCCP ALG | 319](#)
- [Verifying the Security Polices of SIP ALG | 320](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying Source NAT Rule Usage

Purpose

Verify that there is traffic matching the source NAT rule.

Action

From operational mode, enter the `show security nat source rule all` command.

```
user@host> show security nat source rule all
```

```
Total rules: 1
Total referenced IPv4/IPv6 ip-prefixes: 3/0
source NAT rule: phone1          Rule-set: phones
Rule-Id                        : 3
Rule position                   : 2
From zone                       : private
To zone                         : public
Match
  Source addresses              : 10.1.1.3      - 10.1.1.3
  Destination port              : 0              - 0
Action                          : interface
  Persistent NAT type           : N/A
  Persistent NAT mapping type   : address-port-mapping
  Inactivity timeout            : 0
  Max session number            : 0
Translation hits                 : 0
```

Meaning

The Translation hits field shows that, there is no traffic matching the source NAT rule.

Verifying Static NAT Configuration

Purpose

Verify that there is traffic matching the static NAT rule set.

Action

From operational mode, enter the `show security nat static rule phone2` command.

```
user@host> show security nat static rule phone2
```

```
Static NAT rule: phone2          Rule-set: to-proxy
Rule-Id                        : 1
Rule position                  : 1
From zone                      : public
Destination addresses          : 172.16.1.2
Host addresses                 : 10.1.1.4
Netmask                        : 32
Host routing-instance          : N/A
Translation hits                : 0
```

Meaning

The Translation hits field shows that, there is no traffic matching the source NAT rule set.

Verifying SCCP ALG

Purpose

Verify that the SCCP ALG is enabled.

Action

From operational mode, enter the `show security alg status | match sccp` command.

```
user@host> show security alg status | match sccp
```

```
SCCP      : Enabled
```

Meaning

The output shows the SCCP ALG status as follows:

- Enabled—Shows the SCCP ALG is enabled.
- Disabled—Shows the SCCP ALG is disabled.

Verifying the Security Policies of SIP ALG

Purpose

Verify that the static NAT between public zone and private zone is set.

Action

From operational mode, enter the `show security policies` command.

```
user@host> show security policies
```

```
from-zone private to-zone public {
  policy out-pol {
    match {
      source-address any;
      destination-address phone2;
      application junos-sccp;
    }
    then {
      permit;
    }
  }
}
from-zone public to-zone private {
  policy in-pol {
    match {
      source-address phone2;
      destination-address [ cm-tftp_server phone1 ];
      application junos-sccp;
    }
    then {
      permit;
    }
  }
}
```

```

    }
  }
}
from-zone private to-zone private {
  policy tftp-pol {
    match {
      source-address any;
      destination-address any;
      application junos-tftp;
    }
    then {
      permit;
    }
  }
}
}

```

Meaning

The sample output shows that the static NAT between public zone and private zone is set.

Verifying SCCP ALG Configurations

IN THIS SECTION

- [Verifying SCCP ALG | 321](#)
- [Verifying SCCP ALG Calls | 322](#)
- [Verifying SCCP ALG Call Details | 323](#)
- [Verifying SCCP ALG Counters | 325](#)

Verifying SCCP ALG

IN THIS SECTION

- [Purpose | 322](#)

- [Action | 322](#)
- [Meaning | 322](#)

Purpose

Display SCCP verification options.

Action

From the CLI, enter the `show security alg sccp` command.

```
user@host> show security alg sccp ?  
Possible completions:  
calls          Show SCCP calls  
counters       Show SCCP counters
```

Meaning

The output shows a list of all SCCP verification parameters. Verify the following information:

- All SCCP calls
- Counters for all SCCP calls

SEE ALSO

| [Application Layer Gateways User Guide](#)

Verifying SCCP ALG Calls

IN THIS SECTION

- [Purpose | 323](#)
- [Action | 323](#)
- [Meaning | 323](#)

Purpose

Display a list of all SCCP calls

Action

From the CLI, enter the `show security alg sccp calls` command.

```
user@host> show security alg sccp calls
Possible completions:
calls          Show SCCP calls
counters       Show SCCP counters
endpoints      Show SCCP endpoints
```

Meaning

The output shows a list of all SCCP verification parameters. Verify the following information:

- All SCCP calls
- Counters for all SCCP calls
- Information about all SCCP endpoints

Verifying SCCP ALG Call Details

IN THIS SECTION

- [Purpose | 323](#)
- [Action | 324](#)
- [Meaning | 324](#)

Purpose

Display details about all SCCP calls.

Action

From the CLI, enter the `show security alg sccp calls detail` command.

```
user@host> show security alg sccp calls detail
Client IP address: 11.0.102.91
Client zone: 7
Call Manager IP: 13.0.99.226
Conference ID: 16789504
Resource manager group: 2048
SCCP channel information:
  Media transmit channel address (IP address/Port): 0.0.0.0:0
  Media transmit channel translated address (IP address/Port): 0.0.0.0:0
  Media transmit channel pass-through party ID (PPID): 0
  Media transmit channel resource ID: 0
  Media receive channel address (IP address/Port): 11.0.102.91:20060
  Media receive channel translated address (IP address/Port): 25.0.0.1:1032
  Media receive channel pass-through party ID (PPID): 16934451
  Media receive channel resource ID: 8185
  Multimedia transmit channel address (IP address/Port): 0.0.0.0:0
  Multimedia transmit channel translated address (IP address/Port): 0.0.0.0:0
  Multimedia transmit channel pass-through party ID (PPID): 0
  Multimedia transmit channel resource ID: 0
  Multimedia receive channel address (IP address/Port): 0.0.0.0:0
  Multimedia receive channel translated address (IP address/Port): 0.0.0.0:0
  Multimedia receive channel pass-through party ID (PPID): 0
  Multimedia receive channel resource ID: 0
Total number of calls = 1
```

Meaning

The output shows a list of all SCCP verification parameters. Verify the following information:

- Client zone
- Call Manager IP address: 13.0.99.226
- Conference ID
- Resource manager group
- SCCP channel information

- Total number of calls

Verifying SCCP ALG Counters

IN THIS SECTION

- Purpose | 325
- Action | 325
- Meaning | 326

Purpose

Display a list of all SCCP counters

Action

From the J-Web interface, select **Monitor>ALGs>SCCP>Counters**. Alternatively, from the CLI, enter the `show security alg sccp counters` command.

```
user@host> show security alg sccp ounters
```

SCCP call statistics:

Active client sessions	: 0
Active calls	: 0
Total calls	: 0
Packets received	: 0
PDUs processed	: 0
Current call rate	: 0

Error counters:

Packets dropped	: 0
Decode errors	: 0
Protocol errors	: 0
Address translation errors	: 0
Policy lookup errors	: 0
Unknown PDUs	: 0
Maximum calls exceeded	: 0
Maximum call rate exceeded	: 0
Initialization errors	: 0
Internal errors	: 0
Nonspecific error	: 0

```

No active calls to delete          : 0
No active client sessions to delete : 0
Session cookie create errors      : 0
Invalid NAT cookie detected       : 0

```

Meaning

The output shows a list of all SCCP verification parameters. Verify the following information:

- SCCP call statistics
- Error counters

Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
12.1X46-D10	Starting in Junos OS Release 12.1X46-D10 and Junos OS Release 17.3R1, the SCCP ALG supports SCCP versions 16, 17, and 20 and several SCCP messages have been updated with a new format.

RELATED DOCUMENTATION

[Understanding VoIP ALG Types | 188](#)

[VoIP DSCP Rewrite Rules | 189](#)

[H.323 ALG | 192](#)

SIP ALG

SUMMARY

The Session Initiation Protocol (SIP) is a signaling protocol for initiating, modifying, and terminating multimedia sessions over the internet. SIP supports single-media and multi-media sessions.

IN THIS SECTION

● [Understanding the SIP ALG | 327](#)

- [Understanding SIP ALG Hold Resources | 337](#)
- [Understanding the SIP ALG and NAT | 338](#)
- [Example: Setting SIP ALG Call Duration and Timeouts | 352](#)
- [Example: Configuring SIP ALG DoS Attack Protection | 354](#)
- [Example: Allowing Unknown SIP ALG Message Types | 356](#)
- [Example: Configuring Interface Source NAT for Incoming SIP Calls | 358](#)
- [Example: Decreasing Network Complexity by Configuring a Source NAT Pool for Incoming SIP Calls | 367](#)
- [Example: Configuring Static NAT for Incoming SIP Calls | 380](#)
- [Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone | 391](#)
- [Example: Configuring a Three-Zone SIP ALG and NAT Scenario | 403](#)

Understanding the SIP ALG

IN THIS SECTION

- [SIP ALG Operation | 328](#)
- [SDP Session Descriptions | 329](#)
- [Pinhole Creation | 330](#)

Session Initiation Protocol (SIP) is an Internet Engineering Task Force (IETF)-standard protocol for initiating, modifying, and terminating multimedia sessions over the Internet. Such sessions might include conferencing, telephony, or multimedia, with features such as instant messaging and application-level mobility in network environments.

Junos OS supports SIP as a service, allowing and denying it based on a policy that you configure. SIP is a predefined service in Junos OS and uses port 5060 as the destination port.

One of SIP's function is to distribute session-description information, and during the session, to negotiate and modify the parameters of the session. SIP is also used to terminate a multimedia session, signal a call establishment, provide failure indication, and provide methods for endpoint registration.

Session-description information is included in INVITE and 200-OK messages or 200-OK and ACK messages and indicates the multimedia type of the session; for example, whether it is voice or video. Although SIP can use different description protocols to describe the session, the Juniper Networks SIP Application Layer Gateway (ALG) supports only the Session Description Protocol (SDP).

SDP provides information that a system can use to join a multimedia session. SDP might include information such as IP addresses, port numbers, times, and dates. Note that the IP address and port number in the SDP header (the c= and m= fields, respectively) are the address and port where the client wants to receive the media streams and not the IP address and port number from which the SIP request originates (although they can be the same).

SIP messages consist of requests from a client to a server and responses to the requests from a server to a client with the purpose of establishing a session (or a call). A user agent (UA) is an application that runs at the endpoints of the call and consists of two parts:

- user agent client (UAC), which sends SIP requests on behalf of the user
- user agent server (UAS), which listens to the responses and notifies the user when they arrive

UAC and UAS are defined in relation to the role a particular agent is playing in a negotiation.

Examples of UAs are SIP proxy servers and phones.

This topic contains the following sections:

SIP ALG Operation

There are two types of SIP traffic, the signaling and the media stream. SIP signaling traffic consists of request and response messages between client and server and uses transport protocols such as UDP or TCP. The media stream carries the data (audio data, for example) using transport protocols.

Starting in Junos OS Release 12.3X48-D25 and Junos OS Release 17.3R1, the SIP ALG supports TCP. TCP support over the SIP ALG reduces traffic to the server by eliminating the need to reregister or refresh the server frequently.

By default, Junos OS supports SIP signaling messages on port 5060. You can configure the port by creating a policy that permits SIP service, and the software filters SIP signaling traffic like any other type of traffic, permitting or denying it. The media stream, however, uses dynamically assigned port numbers that can change several times during the course of a call. Without fixed ports, it is insecure to create a static policy to control media traffic. In this case, the device invokes the SIP ALG. The device transport

ports used for the media sessions are not known in advance; however, the ports used for the SIP negotiation are well-known (or predefined). The ALG registers interest in packets from the control session, which it can easily distinguish from the other packets, and inspects the negotiation for the transport information used for the media session (both IP addresses and ports).



NOTE: The SIP ALG creates a pinhole when it determines a matching IP, port, transport address, and protocol, which are identified with whatever information is known at the time when the pinhole is opened.

The SIP ALG monitors SIP transactions and dynamically creates and manages pinholes based on the information it extracts from these transactions. The Juniper Networks SIP ALG supports all SIP methods and responses. You can allow SIP transactions to traverse the Juniper Networks firewall by creating a static policy that permits SIP service. If the policy is configured to inspect SIP traffic (or, more appropriately, if the policy sends some traffic to the SIP ALG for inspection), the allowed actions are to permit the traffic (in which case the appropriate pinholes are opened) or to deny the traffic.

The SIP ALG intercepts SIP messages that contain SDP and, using a parser, extracts the information it requires to create pinholes. The SIP ALG examines the SDP portion of the packet, and a parser extracts information such as IP addresses and port numbers, which the SIP ALG records in a pinhole table. The SIP ALG uses the IP addresses and port numbers recorded in the pinhole table to open pinholes and allow media streams to traverse the device.



NOTE: When the device is performing NAT, the transport addresses that the UAs employ are incorrect. The SIP ALG modifies the transport addresses based on the translated ports and addresses allocated by the device translating network addresses. When SDP is encrypted, the device cannot either extract or modify the contents of the message and therefore cannot correct the transport addresses. To provide a workaround, the STUN protocol has been deployed (requiring NAT devices to do some form of cone-NAT), which allows the clients to determine the translated addresses and use those newly discovered addresses in the SDP messages.

NEC SIP products are conditionally supported.

SDP Session Descriptions

An SDP session description is a well-defined format for conveying sufficient information to discover and participate in a multimedia session. A session is described by a series of attribute/value pairs, one per line. The attribute names are single characters, followed by =, and a value. Optional values are specified with =*. Values are either an ASCII string, or a sequence of specific types separated by spaces. Attribute names are only unique within the associated syntactic construct, such as within the session, time, or media only.



NOTE: In the SDP session description, the media-level information begins with the m= field.

Of the many fields in the SDP description, two are particularly useful to the SIP ALG because they contain Transport Layer information.

- c= for connection information

This field can appear at the session or media level. It appears in this format:

c=<network-type><address-type><connection-address>

Junos OS supports only "IN" (for Internet) as the network type, "IPv4" as the address type, and a unicast IP address or domain name as the destination (connection) IP address. Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the "IPv6" address type is also supported.

If the destination IP address is a unicast IP address, the SIP ALG creates pinholes using the IP address and port numbers specified in the media description field m=.

- m= for media announcement

This field appears at the media level and contains the description of the media. It appears in this format:

m=<media><port><transport><fmt list>

Currently, Junos OS supports "RTP" as the Application Layer transport protocol. The port number indicates the destination port of the media stream (the origin is allocated by the remote UA). The format list (fmt list) provides information on the Application Layer protocol that the media uses.

The software opens ports only for RTP and Real-Time Control Protocol (RTCP). Every RTP session has a corresponding RTCP session. Therefore, whenever a media stream uses RTP, the SIP ALG must reserve ports (create pinholes) for both RTP and RTCP traffic. By default, the port number for RTCP is one higher than the RTP port number.

Pinhole Creation

Each pinhole (one for RTP traffic and the other for RTCP traffic) share the same destination IP address. The IP address comes from the c= field in the SDP session description. Because the c= field can appear in either the session-level or the media-level portion of the SDP session description, the parser determines the IP address based on the following rules (in accordance with SDP conventions):

- First, the SIP ALG parser looks for a c= field containing an IP address in the media level. If there is such a field, the parser extracts that IP address, and the SIP ALG uses that address to create a pinhole for the media.

- If there is no c= field in the media level, the SIP ALG parser extracts the IP address from the c= field in the session level, and the SIP ALG uses that IP address to create a pinhole for the media. If the session description does not contain a c= field in either level, this indicates an error in the protocol stack, and the device drops the packet and logs the event.

The SIP ALG also opens pinholes for signal traffic. These signal pinholes are useful after the previous signal session timeout, and they are also useful for the signal traffic sent to a third-party address that does not match with the previous signal session. The SIP ALG signal pinholes never age out, unlike RTP or RTCP pinholes, where only the destination IP and destination port are specified.

The SIP ALG opens signal pinholes for following headers, if needed:

- VIA
- CONTACT
- ROUTE
- RECORD-ROUTE

The SIP ALG needs the following information to create a pinhole. This information either comes from the SDP session description or from the SIP headers (as listed above).

- Protocol—UDP or TCP.
- Source IP—Unknown.
- Source port—Unknown.
- Destination IP—The parser extracts the destination IP address from the c= field at the media or session level.
- Destination port—The parser extracts the destination port number for RTP from the m= field in the media level and calculates the destination port number for RTCP using the following formula:

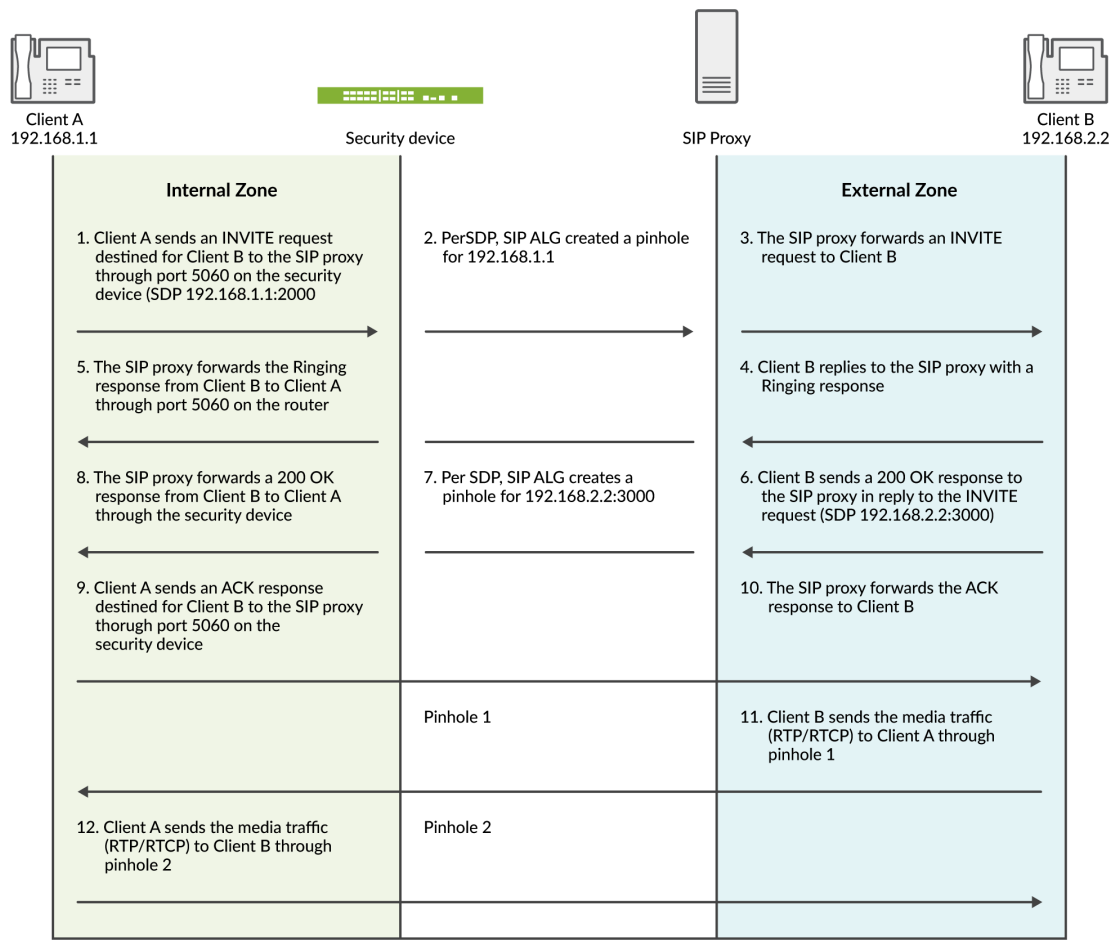
RTP port number + one

- Lifetime—This value indicates the length of time (in seconds) during which a pinhole is open to allow a packet through. A packet must go through the pinhole before the lifetime expires. When the lifetime expires, the SIP ALG removes the pinhole.

When a packet goes through the pinhole within the lifetime period, immediately afterwards the SIP ALG removes the pinhole for the direction from which the packet came.

[Figure 27 on page 332](#) describes a call setup between two SIP clients and how the SIP ALG creates pinholes to allow RTP and RTCP traffic. The illustration assumes that the device has a policy that permits SIP, thus opening port 5060 for SIP signaling messages.

Figure 27: SIP ALG Call Setup



NOTE: The SIP ALG does not create pinholes for RTP and RTCP traffic when the destination IP address is 0.0.0.0, which indicates that the session is on hold. To put a session on hold during a telephone communication, for example, Client A sends Client B a SIP message in which the destination IP address is 0.0.0.0. Doing so indicates to Client B that it should not send any media until further notice. If Client B sends media anyway, the device drops the packets.

Understanding IPv6 Support for SIP ALG

IPv6 is supported on the SIP ALG along with NAT-PT mode and NAT64 address translation.

The SIP ALG processes the IPv6 address in the same way it processes the IPv4 address for updating the payload if NAT is configured and opening pinholes for future traffic.

Special processing occurs for the following formats:

- **IPv6 in SIP URIs**—The SIP URI looks the same as a URI with IPv4 addresses. As in all URIs, an IPv6 address is enclosed in square brackets. The IPv6 address blocks are separated by colons. In many notations, a colon separates the hostname or IP address from the protocol port. To parse the full IPv6 address and separate the port, the address is encapsulated within square brackets
- **IPv6 in SDP**—IPv6 addresses in the Session Description Protocol (SDP) have the IP6 marker.
- The SIP ALG with IPv6 support has the following limitation:
 - When NAT64 with persistent NAT is implemented, the SIP ALG adds the NAT translation to the persistent NAT binding table if NAT is configured on the Address of Record (AOR). Because persistent NAT cannot duplicate the address configured, coexistence of NAT66 and NAT64 configured on the same address is not supported.

Only one binding is created for the same source IP address.

Understanding Scaling Busy Lamp Field Support for the UDP-Based SIP ALG

Busy lamp field (BLF) is a light on an IP phone that indicates whether another extension connected to the same private branch exchange (PBX) is busy or not. You can manually configure the BLF by using a Web interface. When BLF is configured, the phone subscribes to a resource list available on the IP PBX to be notified of status information for other extensions. BLF works through the Session Initiation Protocol (SIP) and uses the SUBSCRIBE and NOTIFY messages. Usually, the phone is the subscriber and the IP PBX is the notifier.

When a phone is registered to the IP PBX, the IP PBX notifies the phone of the state of the resource list. For example, if the resource list is huge, the body of the NOTIFY message will also be huge. Because the SIP ALG supports only 3000-byte SIP messages, it bypasses the huge NOTIFY message. If there are too many instances of BLF in the message body, the payload will not be changed and the gate will not be opened.

Starting with Junos OS Release 12.3X48-D15 and Junos OS Release 17.3R1, the SIP ALG supports 65,000-byte SIP messages on the UDP protocol. In the scaling BLF application, if every instance is around 500 bytes, the SIP ALG supports 100 instances in one SIP UDP message.

BLF support for the UDP-based SIP ALG includes the following features:

- The device can send and receive 65,000-byte SIP messages.
- The SIP ALG can parse the 65,000-byte SIP messages and open the pinhole, if required.
- The SIP ALG regenerates the new jumbo SIP message if NAT is configured and the payload is changed.

Understanding SIP ALG Request Methods

The Session Initiation Protocol (SIP) transaction model includes a number of request and response messages, each of which contains a *method* field that denotes the purpose of the message.

Junos OS supports the following method types and response codes:

- INVITE—A user sends an INVITE request to invite another user to participate in a session. The body of an INVITE request can contain the description of the session.
- ACK—The user from whom the INVITE originated sends an ACK request to confirm reception of the final response to the INVITE request. If the original INVITE request did not contain the session description, the ACK request must include it.
- OPTIONS—The User Agent (UA) obtains information about the capabilities of the SIP proxy. A server responds with information about what methods, session description protocols, and message encoding it supports.
- BYE—A user sends a BYE request to abandon a session. A BYE request from either user automatically terminates the session.
- CANCEL—A user sends a CANCEL request to cancel a pending INVITE request. A CANCEL request has no effect if the SIP server processing the INVITE had sent a final response for the INVITE before it received the CANCEL.
- REGISTER—A user sends a REGISTER request to a SIP registrar server to inform it of the current location of the user. A SIP registrar server records all the information it receives in REGISTER requests and makes this information available to any SIP server attempting to locate a user.
- Info—Used to communicate mid-session signaling information along the signaling path for the call.
- Subscribe—Used to request current state and state updates from a remote node.
- Notify—Sent to inform subscribers of changes in state to which the subscriber has a subscription.
- Refer—Used to refer the recipient (identified by the Request-URI) to a third party by the contact information provided in the request.

For example, if user A in a private network refers user B, in a public network, to user C, who is also in the private network, the SIP Application Layer Gateway (ALG) allocates a new IP address and port number for user C so that user C can be contacted by user B. If user C is registered with a registrar, however, its port mapping is stored in the ALG Network Address Translation (NAT) table and is reused to perform the translation.

- Update—Used to open pinhole for new or updated SDP information. The Via:, From:, To:, Call-ID:, Contact:, Route:, and Record-Route: header fields are modified.

- 1xx, 202, 2xx, 3xx, 4xx, 5xx, 6xx Response Codes—Used to indicate the status of a transaction. Header fields are modified.

SIP ALG Configuration Overview

The Session Initiation Protocol Application Layer Gateway (SIP ALG) is disabled by default on SRX device—it should be enabled using the CLI if required. On other devices, it is enabled by default. To fine-tune SIP ALG operations use the following instructions:

1. Control SIP call activity. For instructions, see ["Example: Setting SIP ALG Call Duration and Timeouts" on page 352](#).
2. Protect the SIP proxy server from denial-of-service (DoS) flood attacks. For instructions, see ["Example: Configuring SIP ALG DoS Attack Protection" on page 354](#).
3. Enable unknown messages to pass when the session is in Network Address Translation (NAT) mode and route mode. For instructions, see ["Example: Allowing Unknown SIP ALG Message Types" on page 356](#).
4. Accommodate proprietary SIP call flows. For instructions, see [Retaining SIP ALG Hold Resources \(CLI Procedure\)](#)

Understanding SIP ALG DoS Attack Protection

The ability of the Session Initiation Protocol (SIP) proxy server to process calls can be impacted by repeat SIP INVITE requests—requests that it initially denied. The denial-of-service (DoS) protection feature enables you to configure the device to monitor INVITE requests and proxy server replies to them. If a reply contains a 3xx, 4xx, or 5xx response code other than 401, 407, 487, and 488 that are not real failure responses, then the request should not be blocked. See ["Understanding the SIP ALG and NAT" on page 338](#). The ALG stores the source IP address of the request and the IP address of the proxy server in a table. Subsequently, the device checks all INVITE requests against this table and, for a configurable number of seconds (the default is 3), discards any packets that match entries in the table. You can configure the device to monitor and deny repeat INVITE requests to all proxy servers, or you can protect a specific proxy server by specifying the destination IP address. SIP attack protection is configured globally.

Understanding SIP ALG Unknown Message Types

This feature enables you to specify how unidentified Session Initiation Protocol (SIP) messages are handled by the device. The default is to drop unknown (unsupported) messages.

We do not recommend permitting unknown messages because they can compromise security. However, in a secure test or production environment, this command can be useful for resolving interoperability issues with disparate vendor equipment. Permitting unknown SIP messages can help you get your

network operational so you can later analyze your voice-over-IP (VoIP) traffic to determine why some messages were being dropped. The unknown SIP message type feature enables you to configure the device to accept SIP traffic containing unknown message types in both Network Address Translation (NAT) mode and route mode.



NOTE: This option applies only to received packets identified as supported VoIP packets. If a packet cannot be identified, it is always dropped. If a packet is identified as a supported protocol and you have configured the device to permit unknown message types, the message is forwarded without processing.

Understanding SIP ALG Call Duration and Timeouts

The call duration and timeout features give you control over Session Initiation Protocol (SIP) call activity and help you to manage network resources.

Typically a call ends when one of the clients sends a BYE or CANCEL request. The SIP Application Layer Gateway (ALG) intercepts the BYE or CANCEL request and removes all media sessions for that call. There could be reasons or problems preventing clients in a call from sending BYE or CANCEL requests, for example, a power failure. In this case, the call might go on indefinitely, consuming resources on the device.

A call can have one or more voice channels. Each voice channel has two sessions (or two media streams), one for Real-Time Transport Protocol (RTP) traffic and one for Real-Time Control Protocol (RTCP) signaling. When managing the sessions, the device considers the sessions in each voice channel as one group. Timeouts and call duration settings apply to a group as opposed to each session.

The following parameters govern SIP call activity:

- **inactive-media-timeout**—This parameter indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the temporary openings (pinholes) in the firewall the SIP ALG opened for media are closed. The default setting is 120 seconds, and the range is 10 through 2550 seconds. Note that upon timeout, resources for media (sessions and pinholes) are removed and SIP calls on the device will also be terminated if all the media resources of this call are removed.
- **maximum-call-duration**—This parameter sets the absolute maximum length of a call. When a call exceeds this parameter setting, the SIP ALG tears down the call and releases the media sessions. The default setting is 720 minutes, and the range is 3 through 720 minutes.
- **t1-interval**—This parameter specifies the roundtrip time estimate, in seconds, of a transaction between endpoints. The default is 500 milliseconds. Because many SIP timers scale with the t1-interval (as described in RFC 3261), when you change the value of the t1-interval timer, those SIP timers also are adjusted.

- **t4-interval**—This parameter specifies the maximum time a message remains in the network. The default is 5 seconds and the range is 5 through 10 seconds. Because many SIP timers scale with the t4-interval (as described in RFC 3261), when you change the value of the t4-interval timer, those SIP timers also are adjusted.
- **c-timeout**—This parameter specifies the INVITE transaction timeout at the proxy, in minutes; the default is 3. Because the SIP ALG is in the middle, instead of using the INVITE transaction timer value B (which is $(64 * T1) = 32$ seconds), the SIP ALG gets its timer value from the proxy.

Understanding SIP ALG Hold Resources

IN THIS SECTION

- [Retaining SIP ALG Hold Resources \(CLI Procedure\) | 337](#)

When a user puts a call on hold, the Session Initiation Protocol Application Layer Gateway (SIP ALG) releases Session Description Protocol (SDP) media resources, such as pinholes and translation contexts. When the user resumes the call, an INVITE request message negotiates a new SDP offer and answer and the SIP ALG reallocates resources for the media stream. This can result in new translated IP address and port numbers for the media description even when the media description is the same as the previous description. This is compliant with *RFC 3264 An Offer/Answer Model with the Session Description Protocol (SDP)*.

Some proprietary SIP implementations have designed call flows so that the User Agent (UA) module ignores the new SDP INVITE offer and continues to use the SDP offer of the previous negotiation. To accommodate this functionality, you must configure the device to retain SDP media resources when a call is put on hold for reuse when the call is resumed.

Retaining SIP ALG Hold Resources (CLI Procedure)

To accommodate proprietary SIP call flows:

```
user@host# set security alg sip retain-hold-resource
```

Understanding the SIP ALG and NAT

IN THIS SECTION

- [Outgoing Calls | 339](#)
- [Incoming Calls | 339](#)
- [Forwarded Calls | 340](#)
- [Call Termination | 340](#)
- [Call Re-INVITE Messages | 340](#)
- [Call Session Timers | 340](#)
- [Call Cancellation | 341](#)
- [Forking | 341](#)
- [SIP Messages | 341](#)
- [SIP Headers | 341](#)
- [SIP Body | 344](#)
- [SIP NAT Scenario | 345](#)
- [Classes of SIP Responses | 347](#)
- [NAT Mode in Pure IPv6 Mode \(NAT66\) for SIP IPv6 ALG | 349](#)
- [NAT-PT | 349](#)
- [NAT64 | 349](#)
- [STUN and SIP ALG | 350](#)

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

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

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

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

When a message containing SDP information arrives, the ALG allocates ports and creates a NAT mapping between them and the ports in the SDP. Because the SDP requires sequential ports for the Real-Time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP) channels, the ALG provides consecutive even-odd ports. If it is unable to find a pair of ports, it discards the SIP message.

IPv6 is supported on the SIP ALG along with NAT-PT mode and NAT64 address translation.

This topic contains the following sections:

Outgoing Calls

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

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

Incoming Calls

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

The ALG examines the SIP request message (initially an INVITE) and, based on information in the SDP, opens gates for outgoing media. When a 200 OK response message arrives, the SIP ALG performs NAT

on the IP addresses and ports and opens pinholes in the outbound direction. (The opened gates have a short time-to-live, and they time out if a 200 OK response message is not received quickly.)

When a 200 OK response arrives, the SIP proxy examines the SDP information and reads the IP addresses and port numbers for each media session. The SIP ALG on the device performs NAT on the addresses and port numbers, opens pinholes for outbound traffic, and refreshes the timeout for gates in the inbound direction.

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

Forwarded Calls

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

Call Termination

The BYE message terminates a call. When the device receives a BYE message, it translates the header fields just as it does for any other message. But because a BYE message must be acknowledged by the receiver with a 200 OK, the ALG delays call teardown for 5 seconds to allow time for transmission of the 200 OK.

Call Re-INVITE Messages

Re-INVITE messages add new media sessions to a call and remove existing media sessions. When new media sessions are added to a call, new pinholes are opened in the firewall and new address bindings are created. The process is identical to the original call setup. When all the media sessions or media pinholes are removed from a call, the call is removed when a BYE message is received.

Call Session Timers

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

- End systems crash during a call and a BYE message is not received.

- Malicious users never send a BYE in an attempt to attack a SIP ALG.
- Poor implementations of SIP proxy fail to process Record-Route and never send a BYE message.
- Network failures prevent a BYE message from being received.

Call Cancellation

Either party can cancel a call by sending a CANCEL message. Upon receiving a CANCEL message, the SIP ALG closes pinholes through the firewall—if any have been opened—and releases address bindings. Before releasing the resources, the ALG delays the control channel age-out for approximately 5 seconds to allow time for the final 200 OK to pass through. The call is terminated when the 5-second timeout expires, regardless of whether a 487 or non-200 response arrives.

Forking

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

SIP Messages

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

SIP Headers

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

```
INVITE bob@10.150.20.5 SIP/2.0
Via: SIP/2.0/UDP 10.150.20.3:5434
From: alice@10.150.20.3
To: bob@10.150.20.5
Call-ID: a12abcde@10.150.20.3
Contact: alice@10.150.20.3:5434
```

Route: <sip:netscreen@10.150.20.3:5060>
 Record-Route: <sip:netscreen@10.150.20.3:5060>

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

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

Table 19 on page 342 shows how NAT is performed in each of these cases. Note that for several of the header fields the ALG determine more than just whether the messages comes from inside or outside the network. It must also determine what client initiated the call, and whether the message is a request or response.

Table 19: Requesting Messages with NAT Table

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

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

Call-ID:	None
Via:	Replace ALG address with local address
Request-URI:	N/A
Contact:	None
Record-Route:	Replace ALG address with local address
Route:	Replace ALG address with local address

SIP Body

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

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

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

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

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

Junos OS supports up to 6 SDP channels negotiated for each direction, for a total of 12 channels per call. For more information, see ["Understanding the SIP ALG" on page 327](#).

SIP NAT Scenario

[Figure 28 on page 346](#) and [Figure 29 on page 347](#) show a SIP call INVITE and 200 OK. In [Figure 28 on page 346](#), ph1 sends a SIP INVITE message to ph2. Note how the IP addresses in the header fields—shown in bold font—are translated by the device.

The SDP section of the INVITE message indicates where the caller is willing to receive media. Note that the Media Pinhole contains two port numbers, 52002 and 52003, for RTCP and RTP. The Via/Contact Pinhole provides port number 5060 for SIP signaling.

Observe how, in the 200 OK response message in [Figure 29 on page 347](#), the translations performed in the INVITE message are reversed. The IP addresses in this message, being public, are not translated, but gates are opened to allow the media stream access to the private network.

Figure 28: SIP NAT Scenario 1

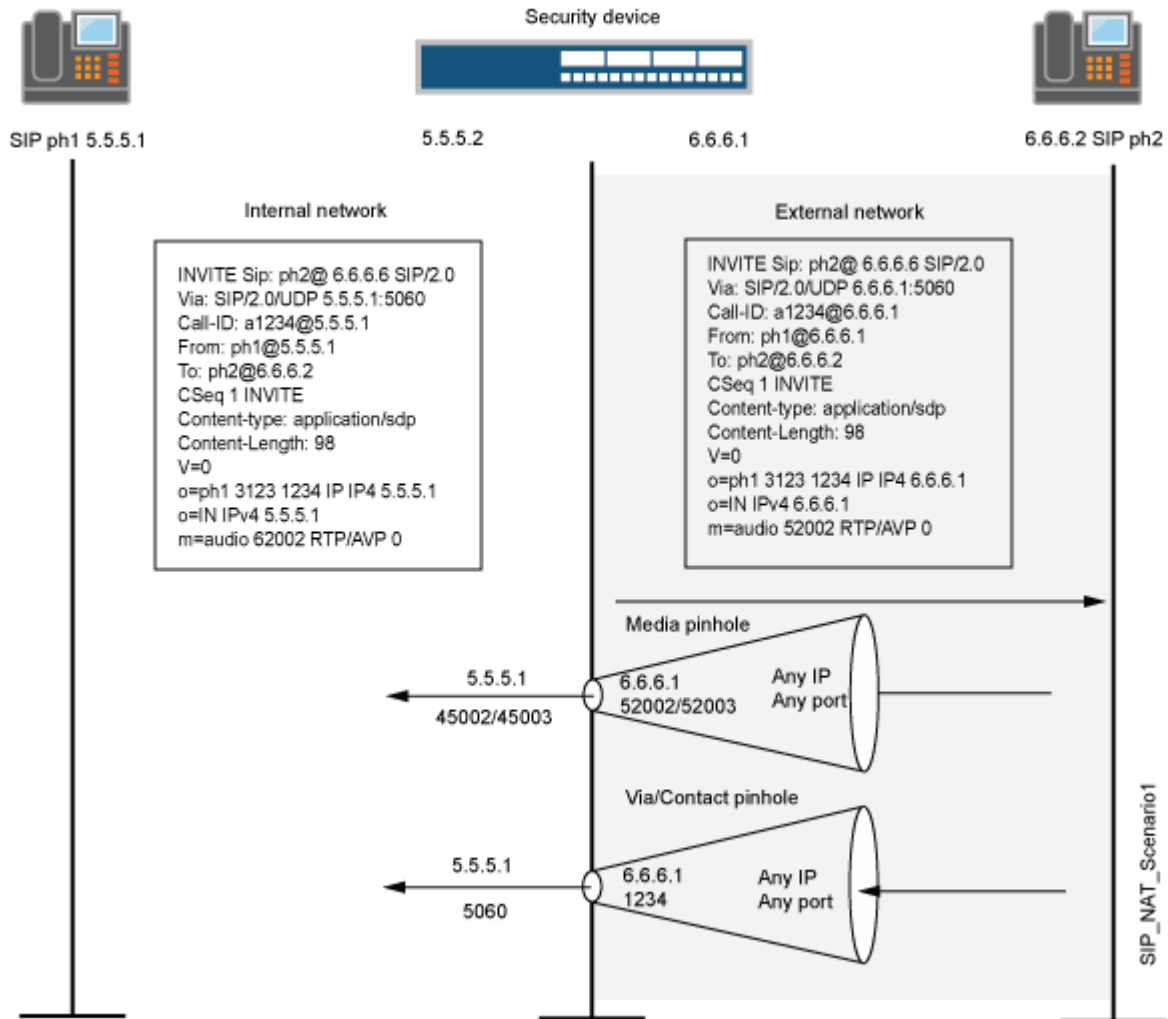
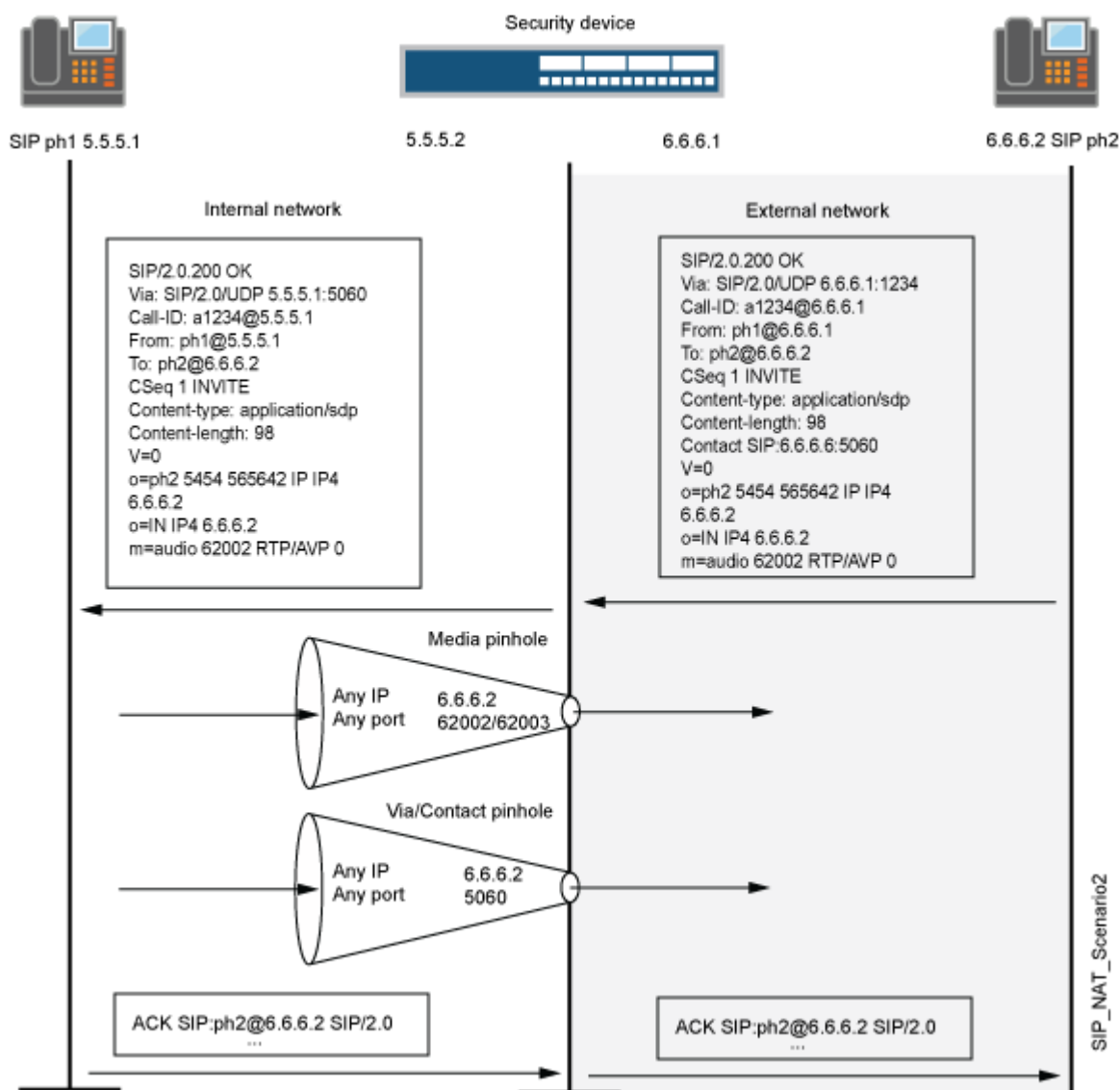


Figure 29: SIP NAT Scenario 2



Classes of SIP Responses

SIP responses provide status information about SIP transactions and include a response code and a reason phrase. SIP responses are grouped into the following classes:

- **Informational (100 to 199)**—Request received, continuing to process the request.
- **Success (200 to 299)**—Action successfully received, understood, and accepted.

- Redirection (300 to 399)—Further action required to complete the request.
- Client Error (400 to 499)—Request contains bad syntax or cannot be fulfilled at this server.
- Server Error (500 to 599)—Server failed to fulfill an apparently valid request.
- Global Failure (600 to 699)—Request cannot be fulfilled at any server.

Table 20 on page 348 provides a complete list of current SIP responses.

Table 20: SIP Responses

Informational	100 Trying	180 Ringing	181 Call is being forwarded
	182 Queued	183 Session progress	
Success	200 OK	202 Accepted	
Redirection	300 Multiple choices	301 Moved permanently	302 Moved temporarily
	305 Use proxy	380 Alternative service	
Client Error	400 Bad request	401 Unauthorized	402 Payment required
	403 Forbidden	404 Not found	405 Method not allowed
	406 Not acceptable	407 Proxy authentication required	408 Request time-out
	409 Conflict	410 Gone	411 Length required
	413 Request entity too large	414 Request URL too large	415 Unsupported media type
	420 Bad extension	480 Temporarily not available	481 Call leg/transaction does not exist
	482 Loop detected	483 Too many hops	484 Address incomplete

	485 Ambiguous	486 Busy here	487 Request canceled
	488 Not acceptable here		
Server Error	500 Server internal error	501 Not implemented	502 Bad gateway
	502 Service unavailable	504 Gateway time-out	505 SIP version not supported
Global Failure	600 Busy everywhere	603 Decline	604 Does not exist anywhere
	606 Not acceptable		

NAT Mode in Pure IPv6 Mode (NAT66) for SIP IPv6 ALG

The SIP IPv6 ALG supports NAT66 just like NAT44. NAT66 (IPv6 NAT) provides source NAT and static NAT functions similar to NAT44 (IPv4 NAT).

NAT-PT

Network Address Translation Protocol Translation (NAT-PT) (RFC 2766) is a protocol translation mechanism that allows communication between IPv6-only and IPv4-only nodes through protocol-independent translation of IPv4 and IPv6 datagrams, requiring no state information for the session.

NAT-PT is implemented by normal NAT from IPv6 address to IPv4 address and vice versa. The SIP ALG processes those address translations in the payload just as the addresses are processed in normal NAT.

NAT-PT binds the addresses in the IPv6 network with addresses in the IPv4 network and vice versa to provide transparent routing for the datagrams traversing between address realms.

The main advantage of NAT-PT is that the end devices and networks can run either IPv4 addresses or IPv6 addresses and traffic can be started from any side.

NAT64

NAT64 is a mechanism to allow IPv6 hosts to communicate with IPv4 servers. NAT64 is required to keep the IPv6 to IPv4 address mapping. Such address mapping is either statically configured by the

system administrator (stateless translation), or more frequently, created automatically when the first packet from the IPv6 network reaches NAT64 to be translated (stateful).

NAT64 is implemented on devices by using persistent NAT. When the first SIP request message (first packet should be only from IPv6) transverses the DUT, address binding is created and then the packets can flow in both directions.

The NAT64 mechanism translates IPv6 packets to IPv4 packets and vice versa, which allows IPv6 clients to contact to the IPv4 servers using unicast UDP, TCP, or ICMP. The NAT-PT and NAT64 behavior seems similar, but these mechanisms are implemented differently.

When NAT64 with persistent NAT is implemented, the SIP ALG with IPv6 support adds the NAT translation to the persistent NAT binding table if NAT is configured on the address of record. Because persistent NAT cannot duplicate the address configured, coexistence of NAT66 and NAT64 configured on the same address is not supported.

Only one binding is created for the same source IP address.

STUN and SIP ALG

Session Traversal Utilities for NAT (STUN) is a solution to make VoIP work through NAT and firewall.

Previously STUN worked without the SIP ALG. This means that the SIP ALG was not involved when persistent NAT was configured.

STUN can coexist with the SIP ALG and SIP ALG is involved when persistent NAT is configured.

Understanding Incoming SIP ALG Call Support Using the SIP Registrar and NAT

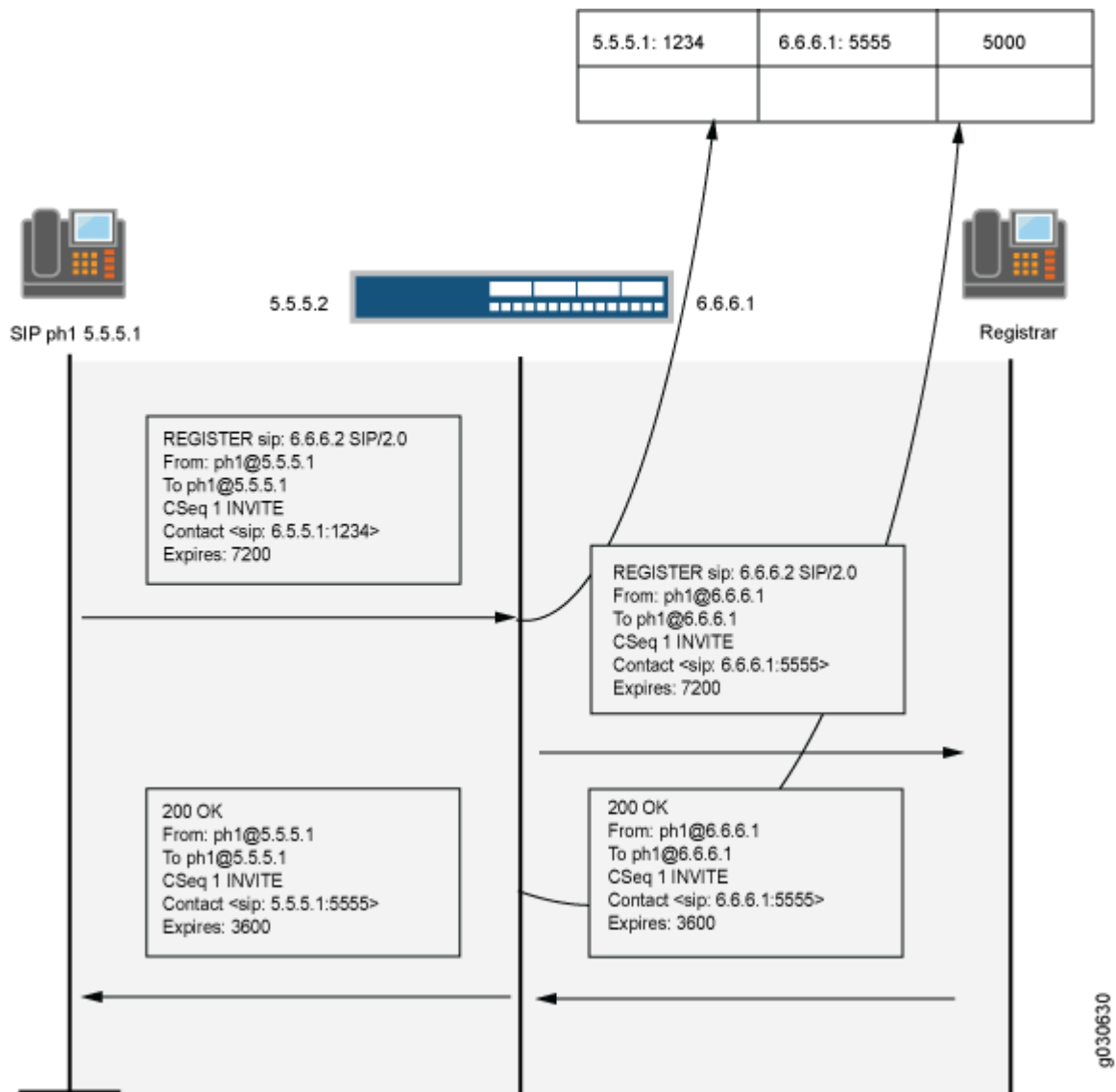
Session Initiation Protocol (SIP) registration provides a discovery capability by which SIP proxies and location servers can identify the location or locations where users want to be contacted. A user registers one or more contact locations by sending a REGISTER message to the registrar. The To and Contact fields in the REGISTER message contain the address-of-record Uniform Resource Identifier (URI) and one or more contact URIs, as shown in [Figure 30 on page 351](#). Registration creates bindings in a location service that associates the address-of-record with the contact address or addresses.

The device monitors outgoing REGISTER messages, performs Network Address Translation (NAT) on these addresses, and stores the information in an Incoming NAT table. Then, when an INVITE message is received from outside the network, the device uses the Incoming NAT table to identify which internal host to route the INVITE message to. You can take advantage of SIP proxy registration service to allow incoming calls by configuring interface source NAT or NAT pools on the egress interface of the device. Interface source NAT is adequate for handling incoming calls in a small office, whereas we recommend setting up source NAT pools for larger networks or an enterprise environment.



NOTE: Incoming call support using interface source NAT or a source NAT pool is supported for SIP and H.323 services only. For incoming calls, Junos OS currently supports UDP and TCP only. Domain name resolution is also currently not supported; therefore, URIs must contain IP addresses, as shown in [Figure 30 on page 351](#).

Figure 30: Using the SIP Registrar



Example: Setting SIP ALG Call Duration and Timeouts

IN THIS SECTION

- Requirements | 352
- Overview | 352
- Configuration | 353
- Verification | 354

This example shows how to set the call duration and the media inactivity timeout.

Requirements

Before you begin, review the call duration and timeout features used to control SIP call activity. See ["Understanding SIP ALG Call Duration and Timeouts" on page 327](#).

Overview

The call duration and inactivity media timeout features help you to conserve network resources and maximize throughput.

The `maximum-call-duration` parameter sets the maximum allowable length of time a call can be active. When the duration is exceeded, the SIP ALG tears down the call and releases the media sessions. The default setting is 720 minutes, and the range is 3 through 720 minutes. This setting also frees up bandwidth in cases where calls fail to properly terminate.

The `inactive-media-timeout` parameter indicates the maximum length of time (in seconds) a call can remain active without any media (RTP or RTCP) traffic within a group. Each time an RTP or RTCP packet occurs within a call, this timeout resets. When the period of inactivity exceeds this setting, the SIP ALG temporary openings (pinholes) for media in the firewall are closed. The default setting is 120 seconds, and the range is 10 through 2550 seconds. Upon timeout, while resources for media (sessions and pinholes) are removed, the call is not terminated.

In this example, the call duration is set to 36000 seconds and the media inactivity timeout is set to 90 seconds.

Configuration

IN THIS SECTION

- [Procedure](#) | 353

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To set the SIP ALG call duration and the media inactivity timeout:

1. Select **Configure** > **Security** > **ALG**.
2. Select the **SIP** tab.
3. In the Maximum call duration field, type 600.
4. In the Inactive media timeout field, enter 90.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options** > **Commit**.

Step-by-Step Procedure

To set the SIP ALG call duration and the media inactivity timeout:

1. Configure the SIP ALG call duration.

```
[edit]  
user@host# set security alg sip maximum-call-duration 600
```

2. Configure the SIP ALG inactivity media timeout.

```
[edit]  
user@host# set security alg sip inactive-media-timeout 90
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg sip` command.

Example: Configuring SIP ALG DoS Attack Protection

IN THIS SECTION

- Requirements | 354
- Overview | 354
- Configuration | 355
- Verification | 356

This example shows how to configure the DoS attack protection feature.

Requirements

Before you begin, review the DoS attack protection feature used to control SIP call activity. See ["Understanding SIP ALG DoS Attack Protection" on page 327](#).

Overview

The ability of the SIP proxy server to process calls can be impacted by repeat SIP INVITE requests—requests that the server initially denied. The DoS protection feature enables you to configure the device to monitor INVITE requests and proxy server replies to them.

In this example, the device is configured to protect a single SIP proxy server (10.1.1.3) from repeat INVITE requests to which it has already been denied service. Packets are dropped for a period of 5 seconds, after which the device resumes forwarding INVITE requests from those sources.

Configuration

IN THIS SECTION

- [Procedure](#) | 355

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To configure SIP ALG DoS attack protection:

1. Select **Configure>Security>ALG**.
2. Select the **SIP** tab.
3. In the Enable attack protection area, click the **Selected servers** option.
4. In the Destination IP box, enter 10.1.1.3 and click **Add**.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To configure SIP ALG DoS attack protection:

1. Configure the device to protect a single SIP proxy server.

[edit]

```
user@host# set security alg sip application-screen protect deny destination-ip 10.1.1.3
```



NOTE: IPv6 is supported on the SIP ALG along with Network Address Translation Protocol Translation (NAT-PT) mode and NAT64 address translation.

The type of the <destination-ip-address> is changed from IPv4 address to IP prefix to support all kinds of IP addresses, and correspondingly a prefix is supported to allow multiple IP addresses.

2. Configure the device for the deny timeout period.

```
[edit]
user@host# set security alg sip application-screen protect deny timeout 5
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg sip` command.

Example: Allowing Unknown SIP ALG Message Types

IN THIS SECTION

- Requirements | 356
- Overview | 357
- Configuration | 357
- Verification | 358

This example shows how to allow unknown message types.

Requirements

Before you begin, review how unidentified SIP messages are handled by the device. See ["Understanding SIP ALG Unknown Message Types" on page 327](#).

Overview

In this example, you configure the device to allow unknown message types in SIP traffic in both NAT mode and route mode. The default is to drop unknown (unsupported) messages.

Configuration

IN THIS SECTION

- [Procedure](#) | 357

Procedure

GUI Quick Configuration

Step-by-Step Procedure

To allow unknown SIP ALG message types:

1. Select **Configure>Security>ALG**.
2. Select the **SIP** tab.
3. Select the **Enable Permit NAT applied** check box.
4. Select the **Enable Permit routed** check box.
5. Click **OK** to check your configuration and save it as a candidate configuration.
6. If you are done configuring the device, click **Commit Options>Commit**.

Step-by-Step Procedure

To allow unknown SIP ALG message types:

1. Configure the device to allow unknown message types in SIP traffic.

```
[edit]
user@host# set security alg sip application-screen unknown-message permit-nat-applied permit-
routed
```

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

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

Verification

To verify the configuration is working properly, enter the `show security alg sip` command.

Example: Configuring Interface Source NAT for Incoming SIP Calls

IN THIS SECTION

- [Requirements | 358](#)
- [Overview | 358](#)
- [Configuration | 360](#)
- [Verification | 365](#)

This example shows how to configure a source NAT rule on a public zone interface allowing NAT to be used for incoming SIP calls.

Requirements

Before you begin, understand how NAT works with the SIP ALG. See ["Understanding the SIP ALG and NAT" on page 338](#).

Overview

IN THIS SECTION

- [Topology | 359](#)

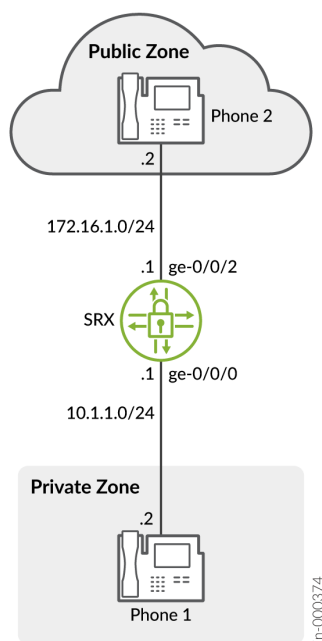
In a two-zone scenario with the SIP proxy server in an external zone, you can use NAT for incoming calls by configuring a source NAT rule on the interface in the public or external zone.

In this example (see [Figure 31 on page 359](#)), phone1 is on the ge-0/0/0 interface in the private zone, and phone2 and the proxy server are on the ge-0/0/2 interface in the public zone. You configure a source NAT rule on the public interface ge-0/0/2.0.

Topology

[Figure 31 on page 359](#) shows source NAT for incoming SIP calls.

Figure 31: Source NAT for Incoming SIP Calls



In this example, after creating zones called private and public and assigning them to interfaces, you configure address books to be used in the source NAT rule set. Then you configure source NAT by defining a rule set called sip-phones and a rule called phone1 that matches any packets from the source address 10.1.1.2/32.

Finally, you create security policies to allow all SIP traffic between the private and public zones.

Configuration

IN THIS SECTION

- [Procedure](#) | 360

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/24
set security zones security-zone private address-book address phone1 10.1.1.2/32
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address proxy 172.16.1.3/32
set security zones security-zone public address-book address phone2 172.16.1.2/32
set security zones security-zone public interfaces ge-0/0/2.0
set security nat source rule-set sip-phones from zone private
set security nat source rule-set sip-phones to zone public
set security nat source rule-set sip-phones rule phone1 match source-address 10.1.1.2/32
set security nat source rule-set sip-phones rule phone1 then source-nat interface
set security policies from-zone private to-zone public policy outgoing match source-address
phone1
set security policies from-zone private to-zone public policy outgoing match destination-address
phone2
set security policies from-zone private to-zone public policy outgoing match destination-address
proxy
set security policies from-zone private to-zone public policy outgoing match application junos-
sip
set security policies from-zone private to-zone public policy outgoing then permit
set security policies from-zone public to-zone private policy incoming match source-address
phone2
set security policies from-zone public to-zone private policy incoming match destination-address
phone1
```

```

set security policies from-zone public to-zone private policy incoming match source-address
proxy
set security policies from-zone public to-zone private policy incoming match application junos-
sip
set security policies from-zone public to-zone private policy incoming then permit

```

Step-by-Step Procedure

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

To configure a source NAT rule on a public zone interface:

1. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/24

```

2. Configure zones and assign them to the interfaces.

```

[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/2.0

```

3. Configure address books and create addresses.

```

[edit security zones]
user@host# set security-zone private address-book address phone1 10.1.1.2/32
user@host# set security-zone public address-book address proxy 172.16.1.3/32
user@host# set security-zone public address-book address phone2 172.16.1.2/32

```

4. Configure a source NAT rule set.

```

[edit security nat source]
user@host# set rule-set sip-phones from zone private
user@host# set rule-set sip-phones to zone public

```

```

user@host# set rule-set sip-phones rule phone1 match source-address 10.1.1.2/32
user@host# set rule-set sip-phones rule phone1 then source-nat interface

```

5. Enable persistent source NAT translation.

```

[edit security nat source]
user@host# set address-persistent

```

6. Configure a security policy to allow outgoing SIP traffic.

```

[edit security policies from-zone private to-zone public policy outgoing]
user@host# set match source-address phone1
user@host# set match destination-address phone2
user@host# set match destination-address proxy
user@host# set match application junos-sip
user@host# set then permit

```

7. Configure a security policy to allow incoming SIP traffic.

```

[edit security policies from-zone public to-zone private policy incoming]
user@host# set match source-address phone2
user@host# set match destination-address phone1
user@host# set match source-address proxy
user@host# set match application junos-sip
user@host# set then permit

```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, `show security policies`, and `show security nat` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

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

```

```

unit 0 {
    family inet {
        address 10.1.1.1/24;
    }
}

ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.16.1.1/24;
        }
    }
}

```

```

[edit]
user@host# show security zones
security-zone private {
    address-book {
        address phone1 10.1.1.2/32;
    }
    interfaces {
        ge-0/0/0.0;
    }
}

security-zone public {
    address-book {
        address proxy 172.16.1.3/32;
        address phone2 172.16.1.2/32;
    }
    interfaces {
        ge-0/0/2.0;
    }
}

[edit]
user@host# show security nat
source {

    rule-set sip-phones {

```



```

    from zone private;
    to zone public;
    rule phone1 {
        match {
            source-address 10.1.1.2/32;
        }
        then {
            source-nat {
                interface;
            }
        }
    }
}
[edit]
user@host# show security policies
from-zone private to-zone public {
    policy outgoing {
        match {
            source-address phone1;
            destination-address [ phone2 proxy ];
            application junos-sip;
        }
        then {
            permit;
        }
    }
}
from-zone public to-zone private {
    policy incoming {
        match {
            source-address [ phone2 proxy ];
            destination-address phone1 ;
            application junos-sip;
        }
        then {
            permit;
        }
    }
}
}

```

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

Verification

IN THIS SECTION

- [Verifying Source NAT Rule Usage | 365](#)
- [Verifying SIP ALG Status | 366](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying Source NAT Rule Usage

Purpose

Verify that there is traffic matching the source NAT rule.

Action

From operational mode, enter the `show security nat source rule all` command. View the Translation hits field to check for traffic that matches the rule.

```
user@host> show security nat source rule all
source NAT rule: phone1      Rule-set: sip-phones
Rule-Id                    : 1
Rule position              : 1
From zone                  : private
To zone                    : public
Match
  Source addresses         : 0.0.0.0      - 255.255.255.255
  Destination port        : 0            - 0
Action                     : interface
Persistent NAT type        : N/A
Persistent NAT mapping type : address-port-mapping
Inactivity timeout        : 0
Max session number        : 0
Translation hits           : 0
Successful sessions       : 0
```

```
Failed sessions      : 0
Number of sessions   : 0
```

Meaning

The Translation hits field shows that, there is no traffic matching the source NAT rule.

Verifying SIP ALG Status

Purpose

Verify that SIP ALG is enabled on your system.

Action

From operational mode, enter the `show security alg status` command.

```
user@host> show security alg status
ALG Status :
  DNS      : Enabled
  FTP      : Enabled
  H323     : Disabled
  MGCP     : Disabled
  MSRPC    : Enabled
  PPTP     : Enabled
  RSH      : Disabled
  RTSP     : Disabled
  SCCP     : Disabled
  SIP      : Enabled
  SQL      : Enabled
  SUNRPC   : Enabled
  TALK     : Enabled
  TFTP     : Enabled
  IKE-ESP  : Disabled
```

Meaning

The output shows the SIP ALG status as follows:

- Enabled—Shows the SIP ALG is enabled.

- Disabled—Shows the SIP ALG is disabled.

Example: Decreasing Network Complexity by Configuring a Source NAT Pool for Incoming SIP Calls

IN THIS SECTION

- Requirements | 367
- Overview | 367
- Configuration | 370
- Verification | 375

This example shows how to decrease network complexity by configuring a source NAT pool on an external interface to enable NAT for incoming SIP calls.

Requirements

Before you begin, understand how NAT works with the SIP ALG. See ["Understanding the SIP ALG and NAT" on page 338](#).

Overview

IN THIS SECTION

- Topology | 368

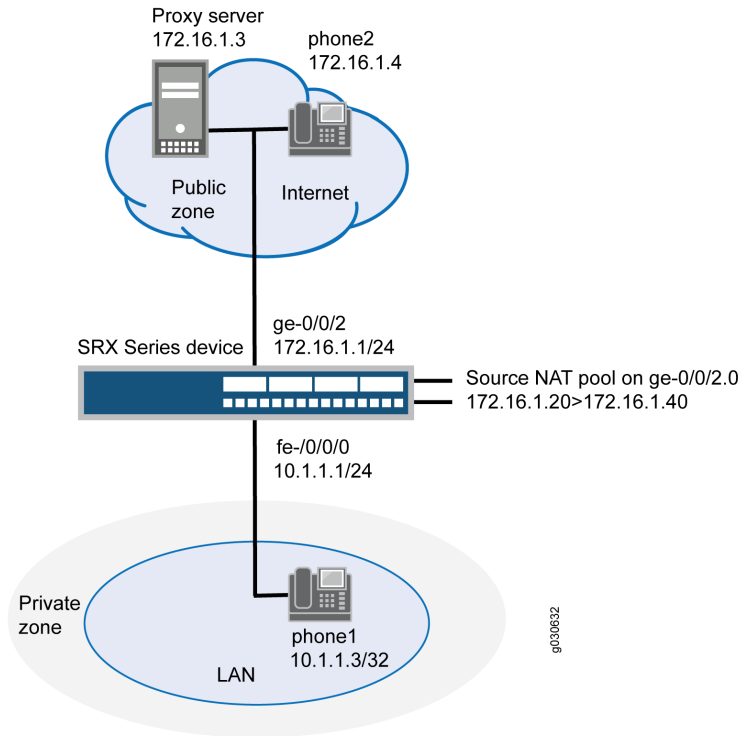
In a two-zone scenario with the SIP proxy server in an external or public zone, you can use NAT for incoming calls by configuring a NAT pool on the interface to the public zone.

In this example (see [Figure 32 on page 369](#)), phone1 is in the private zone, and phone2 and the proxy server are in the public zone. You configure a source NAT pool to do NAT. You also create a policy that permits SIP traffic from the private to the public zone. This enables phone1 in the private zone to register with the proxy server in the public zone, and it also enables incoming calls from the public zone to the private zone.

Topology

[Figure 32 on page 369](#) shows source NAT pool for incoming calls.

Figure 32: Source NAT Pool for Incoming SIP Calls



In this example, you configure source NAT as follows:

- Define source NAT pool called sip-nat-pool to contain the IP address range from 172.16.1.20/32 through 172.16.1.40/32.
- Create a source NAT rule set called sip-nat with a rule sip-r1 to match packets from the private zone to the public zone with the source IP address 10.1.1.3/24. For matching packets, the source address is translated to one of the IP address in sip-nat-pool.
- Configure proxy ARP for the addresses 172.16.1.20/32 through 172.16.1.40/32 on interface ge-0/0/2.0. This allows the system to respond to ARP requests received on the interface for these addresses.

Configuration

IN THIS SECTION

- [Procedure | 370](#)

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/24
set security zones security-zone private address-book address phone1 10.1.1.3/32
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address proxy 172.16.1.3/32
set security zones security-zone public address-book address phone2 172.16.1.4/32
set security zones security-zone public interfaces ge-0/0/2.0
set security nat source pool sip-nat-pool address 172.16.1.20/32 to 172.16.1.40/32
set security nat source address-persistent
set security nat source rule-set sip-nat from zone private
set security nat source rule-set sip-nat to zone public
set security nat source rule-set sip-nat rule sip-r1 match source-address 10.1.1.3/24
```

```

set security nat source rule-set sip-nat rule sip-r1 then source-nat pool sip-nat-pool
set security nat proxy-arp interface ge-0/0/2.0 address 172.16.1.20/32 to 172.16.1.40/32
set security policies from-zone private to-zone public policy outgoing match source-address
phone1
set security policies from-zone private to-zone public policy outgoing match destination-address
any
set security policies from-zone private to-zone public policy outgoing match application junos-
sip
set security policies from-zone private to-zone public policy outgoing then permit
set security policies from-zone public to-zone private policy incoming match source-address
phone2
set security policies from-zone public to-zone private policy incoming match destination-address
phone1
set security policies from-zone public to-zone private policy incoming match application junos-
sip
set security policies from-zone public to-zone private policy incoming then permit

```

Step-by-Step Procedure

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

To configure a source NAT pool for incoming calls:

1. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/24

```

2. Configure zones and assign interfaces to them.

```

[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/2.0

```

3. Configure address books.

```

[edit security zones]
user@host# set security-zone private address-book address phone1 10.1.1.3/32

```



```
user@host# set security-zone public address-book address proxy 172.16.1.3/32
user@host# set security-zone public address-book address phone2 172.16.1.4/32
```

4. Configure a source NAT pool.

```
[edit security nat]
user@host# set source pool sip-nat-pool address 172.16.1.20/32 to 172.16.1.40/32
```

5. Configure a source NAT rule set with a rule.

```
[edit security nat source rule-set sip-nat]
user@host# set from zone private
user@host# set to zone public
user@host# set rule sip-r1 match source-address 10.1.1.3/24
user@host# set rule sip-r1 then source-nat pool sip-nat-pool
```

6. Enable persistent NAT.

```
[edit security nat]
user@host# set source address-persistent
```

7. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/2.0 address 172.16.1.20/32 to 172.16.1.40/32
```

8. Configure a security policy to allow outgoing SIP traffic.

```
[edit security policies from-zone private to-zone public policy outgoing]
set match source-address phone1
set match destination-address any
set match application junos-sip
set then permit
```

9. Configure a security policy to allow incoming SIP traffic.

```
[edit security policies from-zone public to-zone private policy incoming]
set match source-address phone2
set match destination-address phone1
set match application junos-sip
set then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, `show security nat`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 10.1.1.1/24;
      }
    }
  }

  ge-0/0/2 {
    unit 0 {
      family inet {
        address 172.16.1.1/24;
      }
    }
  }
}

[edit]
user@host# show security zones
security-zone private {
  address-book {
    address phone1 10.1.1.3/32;
  }
  interfaces {
    ge-0/0/0.0;
```

```

    }
}
security-zone public {
    address-book {
        address proxy 172.16.1.3/32;
        address phone2 172.16.1.4/32;
    }
    interfaces {
        ge-0/0/2.0;
    }
}
user@host# show security nat
source {

    pool sip-nat-pool {
        address {
            172.16.1.20/32 to 172.16.1.40/32;
        }
    }
    address-persistent;

    rule-set sip-nat {
        from zone private;
        to zone public;
        rule sip-r1 {
            match {
                source-address 10.1.1.3/24;
            }
            then {
                source-nat {
                    pool {
                        sip-nat-pool;
                    }
                }
            }
        }
    }
}
proxy-arp {
    interface ge-0/0/2.0 {
        address {

            172.16.1.20/32 to 172.16.1.40/32;

```

```

    }
  }
}
[edit]
user@host# show security policies
    from-zone private to-zone public {
        policy outgoing {
            match {
                source-address phone1;
                destination-address any;
                application junos-sip;
            }
            then {
                permit;
            }
        }
    }
    from-zone public to-zone private {
        policy incoming {
            match {
                source-address phone2;
                destination-address phone1;
                application junos-sip;
            }
            then {
                permit;
            }
        }
    }
}

```

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

Verification

IN THIS SECTION

- [Verifying Source NAT Pool Usage | 376](#)
- [Verifying Source NAT Rule Usage | 376](#)
- [Verifying SIP ALG Status | 377](#)
- [Verifying the Security Policies of SIP ALG | 378](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying Source NAT Pool Usage

Purpose

Verify that there is traffic using IP addresses from the source NAT pool.

Action

From operational mode, enter the `show security nat source pool all` command.

```
user@host> show security nat source pool all
```

```
Total pools: 1
Pool name      : sip-nat-pool
Pool id       : 4
Routing instance : default
Host address base : 0.0.0.0
Port          : [1024, 63487]
port overloading : 1
Total addresses : 21
Translation hits : 0
Address range   Single Ports  Twin Ports
172.16.1.20 - 172.16.1.40      0           0
```

Meaning

The Translation hits field shows that there is no traffic used by IP addresses from the source NAT pool.

Verifying Source NAT Rule Usage

Purpose

Verify that there is traffic matching the source NAT rule.

Action

From operational mode, enter the `show security nat source rule all` command.

```
user@host> show security nat source rule all
```

```
source NAT rule: sip-r1      Rule-set: sip-nat
Rule-Id                    : 1
Rule position              : 1
From zone                  : private
To zone                    : public
Match
  Source addresses         : 0.0.0.0      - 255.255.255.255
  Destination port         : 0            - 0
Action                     : interface
Persistent NAT type        : N/A
Persistent NAT mapping type : address-port-mapping
Inactivity timeout         : 0
Max session number         : 0
Translation hits           : 0
  Successful sessions      : 0
  Failed sessions         : 0
Number of sessions         : 0
```

Meaning

The Translation hits field shows that, there is no traffic matching the source NAT rule.

Verifying SIP ALG Status

Purpose

Verify that SIP ALG is enabled on your system.

Action

From operational mode, enter the `show security alg status` command.

```
user@host> show security alg status
```

```
ALG Status :  
  DNS      : Enabled  
  FTP      : Enabled  
  H323     : Disabled  
  MGCP     : Disabled  
  MSRPC    : Enabled  
  PPTP     : Enabled  
  RSH      : Disabled  
  RTSP     : Disabled  
  SCCP     : Disabled  
  SIP      : Enabled  
  SQL      : Enabled  
  SUNRPC   : Enabled  
  TALK     : Enabled  
  TFTP     : Enabled  
  IKE-ESP  : Disabled
```

Meaning

The output shows the SIP ALG status as follows:

- Enabled—Shows the SIP ALG is enabled.
- Disabled—Shows the SIP ALG is disabled.

Verifying the Security Policies of SIP ALG

Purpose

Verify that the source NAT between public zone and private zone is set.

Action

From operational mode, enter the `show security policies` command.

```
user@host> show security policies
```

```
from-zone private to-zone public {
  policy outgoing {
    match {
      source-address phone1;
      destination-address any;
      application junos-sip;
    }
    then {
      permit;
    }
  }
}

from-zone public to-zone private {
  policy incoming {
    match {
      source-address phone2;
      destination-address phone1;
      application junos-sip;
    }
    then {
      permit;
    }
  }
}
```

Meaning

The sample output shows that the source NAT between public zone and private zone is set.

Example: Configuring Static NAT for Incoming SIP Calls

IN THIS SECTION

- [Requirements | 380](#)
- [Overview | 380](#)
- [Configuration | 383](#)
- [Verification | 388](#)

This example shows how to configure a static NAT mapping that allows callers in the private zone to register with the proxy server in the public zone.

Requirements

Before you begin, understand how NAT works with the SIP ALG. See ["Understanding the SIP ALG and NAT" on page 338](#).

Overview

IN THIS SECTION

- [Topology | 381](#)

When a SIP proxy server is located in an external or public zone, you can configure static NAT on the public interface to enable callers in the private zone to register with the proxy server.

In this example (see [Figure 33 on page 382](#)), phone1 is on the ge-0/0/0 interface in the private zone, and phone2 and the proxy server are on the ge-0/0/2 interface in the public zone. You create a static NAT rule set called incoming-sip with a rule called phone1 to match packets from the public zone with the destination address 172.16.1.3/32. For matching packets, the destination IP address is translated to the private address 10.1.1.3/32. You also create proxy ARP for the address 172.16.1.3/32 on interface ge-0/0/2.0. This allows the system to respond to ARP requests received on the interface for these addresses. Finally, you create a security policy called incoming that allows SIP traffic from the public zone to the private zone.

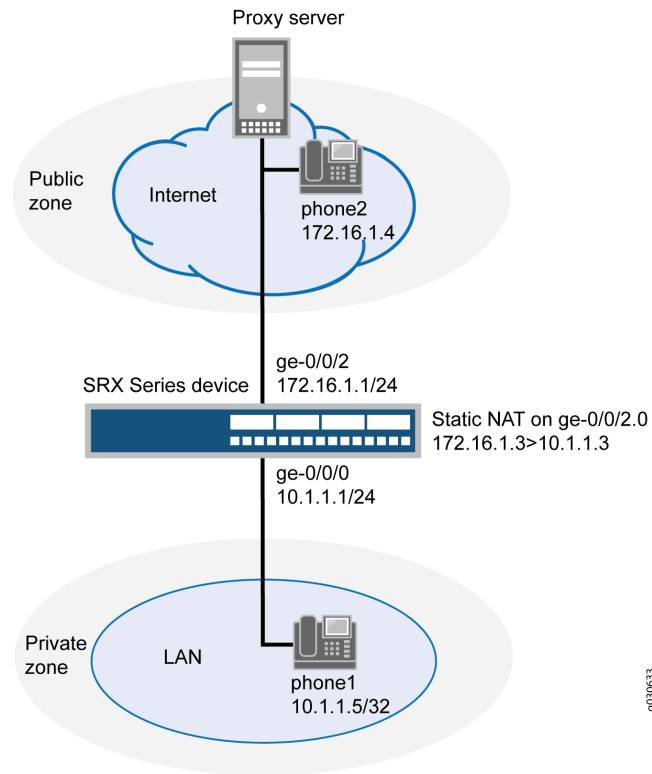


NOTE: When configuring static NAT for incoming SIP calls, make sure to configure one public address for each private address in the private zone.

Topology

[Figure 33 on page 382](#) shows static NAT for incoming calls.

Figure 33: Static NAT for Incoming Calls



Configuration

IN THIS SECTION

- [Procedure](#) | 383

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/24
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone private address-book address phone1 10.1.1.5/32
set security zones security-zone public interfaces ge-0/0/2.0
set security zones security-zone public address-book address proxy 172.16.1.3/32
set security zones security-zone public address-book address phone2 172.16.1.4/32
set security nat static rule-set incoming-sip from zone public
set security nat static rule-set incoming-sip rule phone1 match destination-address
172.16.1.3/32
set security nat static rule-set incoming-sip rule phone1 then static-nat prefix 10.1.1.3/32
set security nat proxy-arp interface ge-0/0/2.0 address 172.16.1.3/32
set security policies from-zone public to-zone private policy incoming match source-address
phone2
set security policies from-zone public to-zone private policy incoming match source-address
proxy
set security policies from-zone public to-zone private policy incoming match destination-address
phone1
set security policies from-zone public to-zone private policy incoming match application junos-
sip
set security policies from-zone public to-zone private policy incoming then permit
set security policies from-zone private to-zone public policy outgoing match source-address
phone1
set security policies from-zone private to-zone public policy outgoing match destination-address
```

```

phone2
set security policies from-zone private to-zone public policy outgoing match destination-address
proxy
set security policies from-zone private to-zone public policy outgoing match application junos-
sip
set security policies from-zone private to-zone public policy outgoing then permit

```

Step-by-Step Procedure

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

To configure static NAT for incoming calls:

1. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/24

```

2. Create security zones.

```

[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/2.0

```

3. Assign addresses to the security zones.

```

[edit security zones]
user@host# set security-zone private address-book address phone1 10.1.1.5/32
user@host# set security-zone public address-book address proxy 172.16.1.3/32
user@host# set security-zone public address-book address phone2 172.16.1.4/32

```

4. Create a static NAT rule set with a rule.

```

[edit security nat static rule-set incoming-sip]
user@host# set from zone public

```

```
user@host# set rule phone1 match destination-address 172.16.1.3/32
user@host# set rule phone1 then static-nat prefix 10.1.1.3/32
```

5. Configure proxy ARP.

```
[edit security nat]
user@host# set proxy-arp interface ge-0/0/2.0 address 172.16.1.3/32
```

6. Define a security policy to allow incoming SIP traffic.

```
[edit security policies from-zone public to-zone private policy incoming]
user@host# set match source-address phone2
user@host# set match source-address proxy
user@host# set match destination-address phone1
user@host# set match application junos-sip
user@host# set then permit
```

7. Define a security policy to allow outgoing SIP traffic.

```
[edit security policies from-zone private to-zone public policy outgoing]
user@host# set match source-address phone1
user@host# set match destination-address phone2
user@host# set match destination-address proxy
user@host# set match application junos-sip
user@host# set then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, `show security nat`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
```

```

    }
}
}

ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.16.1.1/24;
        }
    }
}
}

```

```

[edit]
user@host# show security zones
security-zone private {
    address-book {
        address phone1 10.1.1.5/32;
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone public {
    address-book {
        address proxy 172.16.1.3/32;
        address phone2 172.16.1.4/32;
    }
    interfaces {
        ge-0/0/2.0;
    }
}
}

```

```

[edit]
user@host# show security nat
static {
    rule-set incoming-sip {

        from zone public;
        rule phone1 {
            match {

```



```
}
}
```

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

Verification

IN THIS SECTION

- [Verifying Static NAT Configuration | 388](#)
- [Verifying SIP ALG Status | 389](#)
- [Verifying the Security Policies of SIP ALG | 390](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying Static NAT Configuration

Purpose

Verify that there is traffic matching the static NAT rule set.

Action

From operational mode, enter the `show security nat static rule all` command.

```
user@host> show security nat static rule all
```

```
Static NAT rule: phone1 Rule-set: incoming-sip
Rule-Id : 1
Rule position : 1
From zone : public
Destination addresses : 172.16.1.3
Host addresses : 172.16.1.4
Netmask : 24
Host routing-instance : N/A
Translation hits : 4
Successful sessions : 4
```

```
Failed sessions : 0
Number of sessions : 4
```

Meaning

The Translation hits field shows that there is traffic matching the static NAT rule set.

Verifying SIP ALG Status

Purpose

Verify that SIP ALG is enabled on your system.

Action

From operational mode, enter the `show security alg status` command.

```
user@host> show security alg status
```

```
ALG Status :
  DNS      : Enabled
  FTP      : Enabled
  H323     : Disabled
  MGCP     : Disabled
  MSRPC    : Enabled
  PPTP     : Enabled
  RSH      : Disabled
  RTSP     : Disabled
  SCCP     : Disabled
  SIP      : Enabled
  SQL      : Enabled
  SUNRPC   : Enabled
  TALK     : Enabled
  TFTP     : Enabled
  IKE-ESP  : Disabled
```

Meaning

The output shows the SIP ALG status as follows:

- •Enabled—Shows the SIP ALG is enabled.
- •Disabled—Shows the SIP ALG is disabled.

Verifying the Security Policies of SIP ALG

Purpose

Verify that the static NAT between public zone and private zone is set.

Action

From operational mode, enter the `show security policies` command.

```
user@host> show security policies
```

```
from-zone public to-zone private {
  policy incoming {
    match {
      source-address [ phone2 proxy ];
      destination-address phone1;
      application junos-sip;
    }
    then {
      permit;
    }
  }
}
from-zone private to-zone public {
  policy outgoing {
    match {
      source-address phone1;
      destination-address [ phone2 proxy ];
      application junos-sip;
    }
    then {
      permit;
    }
  }
}
```

Meaning

The sample output shows that the static NAT between public zone and private zone is set.

Example: Configuring the SIP Proxy in the Private Zone and NAT in the Public Zone

IN THIS SECTION

- Requirements | 391
- Overview | 391
- Configuration | 394
- Verification | 399

This example shows how to configure a SIP proxy server in a private zone and static NAT in a public zone to allow callers in the public zone to register with the proxy server.

Requirements

Before you begin, understand how NAT works with the SIP ALG. See ["Understanding the SIP ALG and NAT" on page 338](#).

Overview

IN THIS SECTION

- Topology | 392

With the SIP proxy server in the private zone, you can configure static NAT on the external, or public, interface to allow callers in the public zone to register with the proxy server.

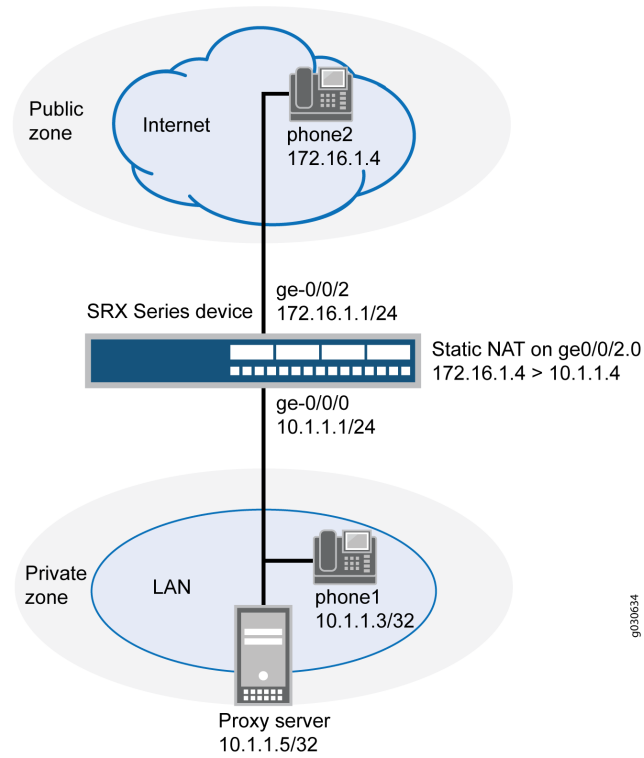
In this example (see [Figure 34 on page 393](#)), phone1 and the SIP proxy server are on the ge-0/0/0 interface in the private zone, and phone2 is on the ge-0/0/2 interface in the public zone. You configure a static NAT rule for the proxy server to allow phone2 to register with the proxy server, and then create a

policy called outgoing that allows SIP traffic from the public to the private zone to enable callers in the public zone to register with the proxy server. You also configure a policy called incoming from the private to the public zone to allow phone1 to call out.

Topology

[Figure 34 on page 393](#) shows configuring SIP proxy in the private zone and NAT in a public zone.

Figure 34: Configuring SIP Proxy in the Private Zone and NAT in a Public Zone



In this example, you configure NAT as follows:

- Configure static NAT on the ge-0/0/2 interface to the proxy server with a rule set called incoming-sip with a rule called proxy to match packets from the public zone with the destination address 172.16.1.2/32. For matching packets, the destination IP address is translated to the private address 10.1.1.5/32.
- Configure a second rule set called sip-phones with a rule called phone1 to enable interface NAT for communication from phone1 to phone2.

Configuration

IN THIS SECTION

- [Procedure | 394](#)

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/24
set interfaces ge-0/0/2 unit 0 proxy-arp
set security zones security-zone private address-book address phone1 10.1.1.3/32
set security zones security-zone private address-book address proxy 10.1.1.5/32
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address phone2 172.16.1.4/32
set security zones security-zone public interfaces ge-0/0/2.0
set security nat source rule-set sip-phones from zone private
set security nat source rule-set sip-phones to zone public
set security nat source rule-set sip-phones rule phone1 match source-address 10.1.1.3/32
set security nat source rule-set sip-phones rule phone1 then source-nat interface
set security nat static rule-set incoming-sip from zone public
set security nat static rule-set incoming-sip rule proxy match destination-address 172.16.1.2/32
set security nat static rule-set incoming-sip rule proxy then static-nat prefix 10.1.1.5/32
```

```

set security nat proxy-arp interface ge-0/0/2.0 address 172.16.1.2/32
set security policies from-zone private to-zone public policy outgoing match source-address any
set security policies from-zone private to-zone public policy outgoing match destination-address
phone2
set security policies from-zone private to-zone public policy outgoing match application junos-
sip
set security policies from-zone private to-zone public policy outgoing then permit
set security policies from-zone public to-zone private policy incoming match source-address
phone2
set security policies from-zone public to-zone private policy incoming match destination-address
proxy
set security policies from-zone public to-zone private policy incoming match application junos-
sip
set security policies from-zone public to-zone private policy incoming then permit

```

Step-by-Step Procedure

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

To configure static NAT for incoming calls:

1. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/24
user@host# set interfaces ge-0/0/2 unit 0 proxy-arp

```

2. Configure security zones.

```

[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/2.0

```

3. Assign addresses to the security zones.

```

[edit security zones]
user@host# set security-zone private address-book address phone1 10.1.1.3/32

```



```

user@host# set security-zone private address-book address proxy 10.1.1.5/32
user@host# set security-zone public address-book address phone2 172.16.1.4/32

```

4. Create a rule set for static NAT and assign a rule to it.

```

[edit security nat static rule-set incoming-sip]
user@host# set from zone public
user@host# set rule proxy match destination-address 172.16.1.2/32
user@host# set rule proxy then static-nat prefix 10.1.1.5/32

```

5. Configure proxy-arp for the 172.16.1.2/32 address.

```

[edit security nat]
user@host# proxy-arp interface ge-0/0/2.0 address 172.16.1.2/32

```

6. Configure the second rule set and assign a rule to it.

```

[edit security nat source rule-set sip-phones]
user@host# set from zone private
user@host# set to zone public
user@host# set rule phone1 match source-address 10.1.1.3/32
user@host# set rule phone1 then source-nat interface

```

7. Configure a security policy for outgoing traffic.

```

[edit security policies from-zone private to-zone public policy outgoing]
user@host# set match source-address any
user@host# set match destination-address phone2
user@host# set match application junos-sip
user@host# set then permit

```

8. Configure a security policy for incoming traffic.

```

[edit security policies from-zone public to-zone private policy incoming]
user@host# set match source-address phone2
user@host# set match destination-address proxy

```

```

user@host# set match application junos-sip
user@host# set then permit

```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, `show security nat`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.1.1.1/24;
        }
    }
}

ge-0/0/2 {
    unit 0 {
        proxy-arp;
        family inet {
            address 172.16.1.1/24;
        }
    }
}
[edit]
user@host# show security zones
security-zone private {
    address-book {
        address phone1 10.1.1.3/32;
        address proxy 10.1.1.5/32;
    }
    interfaces {
        ge-0/0/0.0;
    }
}

security-zone public {
    address-book {
        address phone2 172.16.1.4/32;
    }
}

```

```

    }
    interfaces {
        ge-0/0/2.0;
    }
}
[edit]
user@host# show security nat
source {

    rule-set sip-phones {
        from zone private;
        to zone public;
        rule phone1 {
            match {
                source-address 10.1.1.3/32;
            }
            then {
                source-nat {
                    interface;
                }
            }
        }
    }
}

static {
    rule-set incoming-sip {
        from zone public;
        rule proxy {
            match {
                destination-address 172.16.1.2/32;
            }
            then {
                static-nat prefix 10.1.1.5/32;
            }
        }
    }
}

proxy-arp {
    interface ge-0/0/2.0 {
        address {
            172.16.1.2/32;
        }
    }
}

```

```

    }
[edit]
user@host# show security policies
from-zone private to-zone public {
    policy outgoing {
        match {
            source-address any;
            destination-address phone2;
            application junos-sip;
        }
        then {
            permit;
        }
    }
}
from-zone public to-zone private {
    policy incoming {
        match {
            source-address phone2;
            destination-address proxy;
            application junos-sip;
        }
        then {
            permit;
        }
    }
}
}

```

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

Verification

IN THIS SECTION

- [Verifying Static NAT Configuration | 400](#)
- [Verifying SIP ALG Status | 400](#)
- [Verifying Source NAT Rule | 401](#)
- [Verifying Security Flow Session | 402](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying Static NAT Configuration

Purpose

Verify that there is traffic matching the static NAT rule set.

Action

From operational mode, enter the `show security nat static rule all` command. View the Translation hits field to check for traffic that matches the rule.

```
user@host> show security nat static rule all
Total static-nat rules: 1
Total referenced IPv4/IPv6 ip-prefixes: 2/0
Static NAT rule: proxy          Rule-set: incoming-sip
  Rule-Id                : 2
  Rule position          : 1
  From zone              : public
  Destination addresses  : 172.16.1.2
  Host addresses         : 10.1.1.5
  Netmask                : 32
  Host routing-instance  : N/A
  Translation hits       : 23
    Successful sessions  : 23
    Failed sessions     : 0
  Number of sessions    : 0
```

Meaning

The Translation hits field shows that, there are 23 traffic matching the static NAT rule.

Verifying SIP ALG Status

Purpose

Verify that SIP ALG is enabled on your system.

Action

From operational mode, enter the `show security alg status` command.

```
user@host> show security alg status
```

```
ALG Status :
```

```
DNS      : Enabled
FTP       : Enabled
H323     : Disabled
MGCP     : Disabled
MSRPC    : Enabled
PPTP     : Enabled
RSH      : Disabled
RTSP     : Disabled
SCCP     : Disabled
SIP      : Enabled
SQL      : Enabled
SUNRPC   : Enabled
TALK     : Enabled
TFTP     : Enabled
IKE-ESP  : Disabled
```

Meaning

The output shows the SIP ALG status as follows:

- Enabled—Shows the SIP ALG is enabled.
- Disabled—Shows the SIP ALG is disabled.

Verifying Source NAT Rule

Purpose

Verify that the source NAT rule configuration.

Action

From operational mode, enter the `show security nat source rule all` command.

```
user@host> show security nat source rule all
Total referenced IPv4/IPv6 ip-prefixes: 1/0
source NAT rule: phone1           Rule-set: sip-phones
Rule-Id                          : 1
Rule position                     : 1
From zone                        : private
To zone                          : public
Match
  Source addresses                : 10.1.1.3      - 10.1.1.3
Action                           : interface
  Persistent NAT type             : N/A
  Persistent NAT mapping type    : address-port-mapping
  Inactivity timeout              : 0
  Max session number              : 0
Translation hits                  : 88
  Successful sessions             : 88
  Failed sessions                 : 0
  Number of sessions              : 0
```

Meaning

The Translation hits field shows that, there are 88 traffic matching the source NAT rule.

Verifying Security Flow Session

Purpose

Verify that the NAT translation phone1 to phone2.

Action

From operational mode, enter the `run show security flow session` command.

```
user@host> run show security flow session
Session ID: 169, Policy name: allow-all/4, Timeout: 2, Valid
  In: 10.1.1.3/4 --> 172.16.1.4/52517;icmp, Conn Tag: 0x0, If: ge-0/0/0.0, Pkts: 1, Bytes: 84,
```

```
Out: 172.16.1.4/52517 --> 172.16.1.1/25821;icmp, Conn Tag: 0x0, If: ge-0/0/1.0, Pkts: 1,
Bytes: 84,
```

Meaning

The output displays the NAT translation phone1 to phone2.

Example: Configuring a Three-Zone SIP ALG and NAT Scenario

IN THIS SECTION

- Requirements | 403
- Overview | 403
- Configuration | 406
- Verification | 414

This example shows how to configure a SIP proxy server in a private zone and static NAT in a public zone to allow callers in the public zone to register with the proxy server.

Requirements

Before you begin, understand how NAT works with the SIP ALG. See ["Understanding the SIP ALG and NAT" on page 338](#).

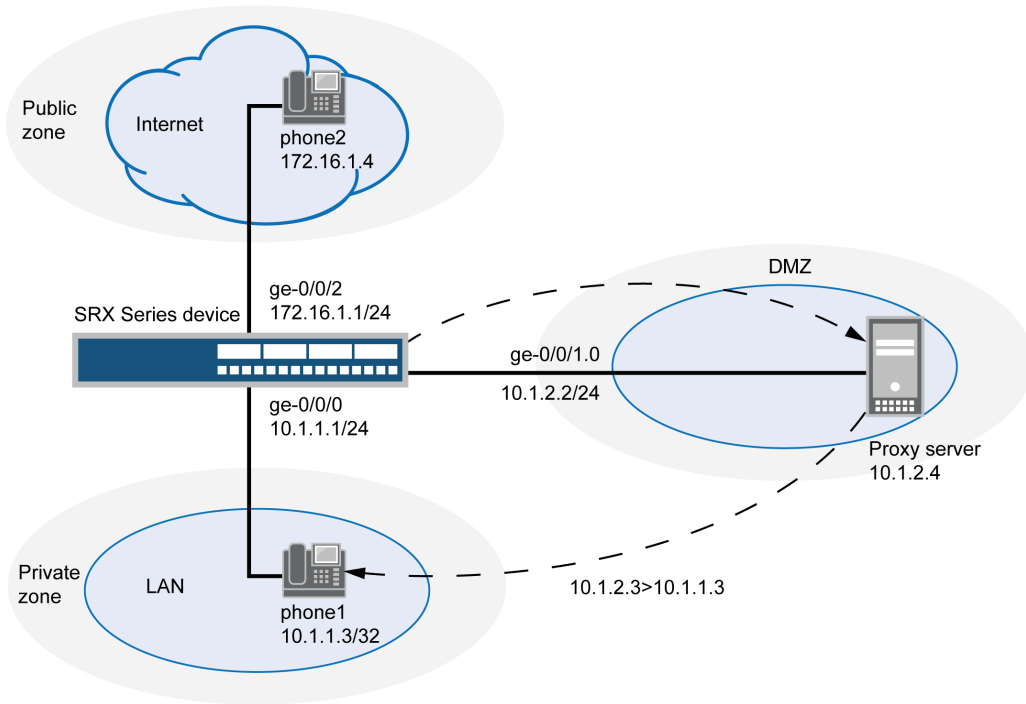
Overview

In a three-zone SIP configuration, the SIP proxy server is typically in a different zone from the calling and called systems. Such a scenario requires additional address and zone configuration, and policies to ensure that all systems have access to each other and to the proxy server.

In this example, phone1 is on the ge-0/0/0.0 interface in the private zone, phone2 is on the ge-0/0/2.0 interface in the public zone, and the proxy server is on the ge-0/0/1.0 interface in the DMZ. You configure static NAT rule for phone1 in the private zone. You then create policies for traffic traversing from the private zone to the DMZ and from the DMZ to the private zone, from the public zone to the DMZ and from the DMZ to the public zone, and from the private zone to the public zone. The arrows in [Figure 35 on page 405](#) show the flow of SIP signaling traffic when phone2 in the public zone places a

call to phone1 in the private zone. After the session is initiated, the data flows directly between phone1 and phone2.

Figure 35: Three-Zone SIP Configuration with Proxy in the DMZ



In this example, you configure NAT as follows:

- Configure a static NAT rule set called incoming-sip with a rule phone1 to match packets from the public zone with the destination address 10.1.2.3/32. For matching packets, the destination IP address is translated to the private address 10.1.1.3/32.
- Configure proxy ARP for the address 10.1.2.3/32 on interface ge-0/0/1.0 allowing the system to respond to ARP requests received on the interface for this address.
- Configure a second rule set called sip-phones with a rule r1 to enable interface NAT for communication from phone1 to the proxy server and from phone1 to phone2.

Configuration

IN THIS SECTION

- [Procedure | 406](#)

Procedure

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the [edit] hierarchy level, and then enter `commit` from configuration mode.

```
set security nat static rule-set sip-phone from zone private
set security nat static rule-set sip-phone from zone public
set security nat static rule-set sip-phone rule phone1 match destination-address 10.1.2.3/32
set security nat static rule-set sip-phone rule phone1 then static-nat prefix 10.1.1.3/32

set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.2/24
set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/24
set security zones security-zone private address-book address phone1 10.1.1.3/32
set security zones security-zone private interfaces ge-0/0/0.0
set security zones security-zone public address-book address phone2 172.16.1.4/32
set security zones security-zone public interfaces ge-0/0/2.0
set security zones security-zone dmz address-book address proxy 10.1.2.4/32
```

```

set security zones security-zone dmz interfaces ge-0/0/1.0
set security nat source rule-set sip-phones from zone private
set security nat source rule-set sip-phones to zone dmz
set security nat source rule-set sip-phones rule r1 match source-address 10.1.1.3/32
set security nat source rule-set sip-phones rule r1 then source-nat interface
set security policies from-zone private to-zone dmz policy private-to-proxy match source-address
phone1
set security policies from-zone private to-zone dmz policy private-to-proxy match destination-
address proxy
set security policies from-zone private to-zone dmz policy private-to-proxy match application
junos-sip
set security policies from-zone private to-zone dmz policy private-to-proxy then permit
set security policies from-zone public to-zone dmz policy public-to-proxy match source-address
phone2
set security policies from-zone public to-zone dmz policy public-to-proxy match destination-
address proxy
set security policies from-zone public to-zone dmz policy public-to-proxy match application
junos-sip
set security policies from-zone public to-zone dmz policy public-to-proxy then permit
set security policies from-zone public to-zone private policy public-to-private match source-
address phone2
set security policies from-zone public to-zone private policy public-to-private match
destination-address phone1
set security policies from-zone public to-zone private policy public-to-private match
application junos-sip
set security policies from-zone public to-zone private policy public-to-private then permit
set security policies from-zone private to-zone public policy private-to-public match source-
address phone1
set security policies from-zone private to-zone public policy private-to-public match
destination-address phone2
set security policies from-zone private to-zone public policy private-to-public match
application junos-sip
set security policies from-zone private to-zone public policy private-to-public then permit
set security policies from-zone dmz to-zone private policy proxy-to-private match source-address
proxy
set security policies from-zone dmz to-zone private policy proxy-to-private match destination-
address phone1
set security policies from-zone dmz to-zone private policy proxy-to-private match application
junos-sip
set security policies from-zone dmz to-zone private policy proxy-to-private then permit
set security policies from-zone dmz to-zone public policy proxy-to-public match source-address
proxy
set security policies from-zone dmz to-zone public policy proxy-to-public match destination-

```

```

address phone2
set security policies from-zone dmz to-zone public policy proxy-to-public match application
junos-sip
set security policies from-zone dmz to-zone public policy proxy-to-public then permit

```

Step-by-Step Procedure

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

To configure a SIP proxy server in a private zone and static NAT in a public zone:

1. Create a rule set for static NAT and assign a rule to it.

```

[edit security nat static rule-set]
user@host# sip-phone from zone private
user@host# sip-phone from zone public
user@host# sip-phone rule phone1 match destination-address 10.1.2.3/32
user@host# phone1 then static-nat prefix 10.1.1.3/32

```

2. Configure interfaces.

```

[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 10.1.1.1/24
user@host# set interfaces ge-0/0/1 unit 0 family inet address 10.1.2.2/24
user@host# set interfaces ge-0/0/2 unit 0 family inet address 172.16.1.1/24

```

3. Configure security zones.

```

[edit security zones]
user@host# set security-zone private interfaces ge-0/0/0.0
user@host# set security-zone public interfaces ge-0/0/2.0
user@host# set security-zone dmz interfaces ge-0/0/1.0

```

4. Assign addresses to the security zones.

```

[edit security zones]
user@host# set security-zone private address-book address phone1 10.1.1.3/32

```

```

user@host# set security-zone public address-book address phone2 172.16.1.4/32
user@host# set security-zone dmz address-book address proxy 10.1.2.4/32

```

5. Configure interface NAT for communication from phone1 to proxy.

```

[edit security nat source rule-set sip-phones]
user@host# set from zone private
user@host# set to zone dmz
user@host# set rule r1 match source-address 10.1.1.3/32
user@host# set rule r1 then source-nat interface

```

6. Configure a security policy to allow traffic from zone private to zone DMZ.

```

[edit security policies from-zone private to-zone dmz policy private-to-proxy]
user@host# set match source-address phone1
user@host# set match destination-address proxy
user@host# set match application junos-sip
user@host# set then permit

```

7. Configure a security policy to allow traffic from zone public to zone DMZ.

```

[edit security policies from-zone public to-zone dmz policy public-to-proxy]
user@host# set match source-address phone2
user@host# set match destination-address proxy
user@host# set match application junos-sip
user@host# set then permit

```

8. Configure a security policy to allow traffic from zone private to zone public.

```

[edit security policies from-zone private to-zone public policy private-to-public]
user@host# set match source-address phone1
user@host# set match destination-address phone2
user@host# set match application junos-sip
user@host# set then permit

```

9. Configure a security policy to allow traffic from zone public to zone private.

```
[edit security policies from-zone public to-zone private policy public-to-private]
user@host# set match source-address phone2
user@host# set match destination-address phone1
user@host# set match application junos-sip
user@host# set then permit
```

10. Configure a security policy to allow traffic from zone DMZ to zone private.

```
[edit security policies from-zone dmz to-zone private policy proxy-to-private]
user@host# set match source-address proxy
user@host# set match destination-address phone1
user@host# set match application junos-sip
user@host# set then permit
```

11. Configure a security policy to allow traffic from zone DMZ to zone public.

```
[edit security policies from-zone dmz to-zone public policy proxy-to-public]
user@host# set match source-address proxy
user@host# set match destination-address phone2
user@host# set match application junos-sip
user@host# set then permit
```

Results

From configuration mode, confirm your configuration by entering the `show interfaces`, `show security zones`, `show security nat`, and `show security policies` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/0 {
  unit 0 {
    family inet {
      address 10.1.1.1/24;
    }
  }
}
```

```

}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.1.2.2/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.16.1.1/24;
        }
    }
}
}

```

```

[edit]
user@host# show security zones
security-zone private {
    address-book {
        address phone1 10.1.1.3/32;
    }
    interfaces {
        ge-0/0/0.0;
    }
}
security-zone public {
    address-book {
        address phone2 172.16.1.4/32;
    }
    interfaces {
        ge-0/0/2.0;
    }
}
security-zone dmz {
    address-book {
        address proxy 10.1.2.4/32;
    }
    interfaces {
        ge-0/0/1.0;
    }
}

```



```
    }
}
```

```
[edit]
user@host# show security nat
static {
    rule-set sip-phone {
        from zone [ private public ];
        rule phone1 {
            match {
                destination-address 10.1.2.3/32;
            }
            then {
                static-nat {
                    prefix {
                        10.1.1.3/32;
                    }
                }
            }
        }
    }
}
source {
    rule-set sip-phones {
        from zone private;
        to zone dmz;
        rule r1 {
            match {
                source-address 10.1.1.3/32;
            }
            then {
                source-nat {
                    interface;
                }
            }
        }
    }
}
proxy-arp {
    interface ge-0/0/1.0 {
```

```

        address {
            10.1.2.3/32;
        }
    }
}

```

```

[edit]
user@host# show security policies
from-zone private to-zone dmz {
    policy private-to-proxy {
        match {
            source-address phone1;
            destination-address proxy;
            application junos-sip;
        }
        then {
            permit;
        }
    }
}
from-zone public to-zone dmz {
    policy public-to-proxy {
        match {
            source-address phone2;
            destination-address proxy;
            application junos-sip;
        }
        then {
            permit;
        }
    }
}
from-zone public to-zone private {
    policy public-to-private {
        match {
            source-address phone2;
            destination-address phone1;
        }
        then {
            permit;
        }
    }
}

```

```

    }
}
from-zone private to-zone public {
    policy private to-zone public {
        match {
            source-address phone1;
            destination-address phone2;
        }
        then {
            permit;
        }
    }
}
from-zone dmz to-zone private {
    policy proxy-to-private {
        match {
            source-address proxy;
            destination-address phone2;
            application junos-sip;
        }
        then {
            permit;
        }
    }
}
}

```

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

Verification

IN THIS SECTION

- [Verifying Source NAT Rule Usage | 415](#)
- [Verifying Static NAT Rule Usage | 415](#)
- [Verifying SIP ALG Status | 416](#)

To confirm that the configuration is working properly, perform these tasks:

Verifying Source NAT Rule Usage

Purpose

Verify that there is traffic matching the source NAT rule.

Action

From operational mode, enter the `show security nat source rule all` command. View the Translation hits field to check for traffic that matches the rule.

```
user@host> show security nat source rule all
source NAT rule: r1      Rule-set: sip-phones
  Rule-Id                : 1
  Rule position          : 1
  From zone              : private
  To zone                : public
  Match
    Source addresses      : 0.0.0.0      - 255.255.255.255
    Destination port      : 0            - 0
  Action                 : interface
    Persistent NAT type   : N/A
    Persistent NAT mapping type : address-port-mapping
    Inactivity timeout    : 0
    Max session number    : 0
  Translation hits       : 0
    Successful sessions   : 0
    Failed sessions       : 0
    Number of sessions    : 0
```

Meaning

The Translation hits field shows that, there is no traffic matching the source NAT rule.

Verifying Static NAT Rule Usage

Purpose

Verify that there is traffic matching the static NAT rule.

Action

From operational mode, enter the `show security nat static rule all` command. View the Translation hits field to check for traffic that matches the rule.

```
user@host> show security nat static rule all
Total static-nat rules: 1
Total referenced IPv4/IPv6 ip-prefixes: 2/0
Static NAT rule: phone1           Rule-set: sip-phone
  Rule-Id                : 1
  Rule position          : 1
  From zone              : private
                        : public
  Destination addresses  : 10.1.2.3
  Host addresses         : 10.1.2.4
  Netmask                : 32
  Host routing-instance  : N/A
  Translation hits       : 127
    Successful sessions  : 127
    Failed sessions     : 0
  Number of sessions    : 0
```

Meaning

The Translation hits field shows that, the traffic matching the static NAT rule.

Verifying SIP ALG Status

Purpose

Verify that SIP ALG is enabled on your system.

Action

From operational mode, enter the `show security alg status` command.

```
user@host> show security alg status
ALG Status :
  DNS      : Enabled
  FTP      : Enabled
```

H323	: Disabled
MGCP	: Disabled
MSRPC	: Enabled
PPTP	: Enabled
RSH	: Disabled
RTSP	: Disabled
SCCP	: Disabled
SIP	: Enabled
SQL	: Enabled
SUNRPC	: Enabled
TALK	: Enabled
TFTP	: Enabled
IKE-ESP	: Disabled

Meaning

The output shows the SIP ALG status as follows:

- Enabled—Shows the SIP ALG is enabled.
- Disabled—Shows the SIP ALG is disabled.

Change History Table

Feature support is determined by the platform and release you are using. Use [Feature Explorer](#) to determine if a feature is supported on your platform.

Release	Description
15.1X49-D40	Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the “IPv6” address type is also supported.
12.3X48-D25	Starting in Junos OS Release 12.3X48-D25 and Junos OS Release 17.3R1, the SIP ALG supports TCP.
12.3X48-D15	Starting with Junos OS Release 12.3X48-D15 and Junos OS Release 17.3R1, the SIP ALG supports 65,000-byte SIP messages on the UDP protocol.

RELATED DOCUMENTATION

Understanding VoIP ALG Types 188
VoIP DSCP Rewrite Rules 189

H.323 ALG | 192

MGCP ALG | 247

4

CHAPTER

Configuration Statements and Operational Commands

IN THIS CHAPTER

- [Junos CLI Reference Overview | 420](#)
-

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Read this guide to learn about the syntax and options that make up the statements and commands. Also understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- [Junos CLI Reference](#)

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- [Configuration Statements](#)
- [Operational Commands](#)