Engineering
Simplicity

Junos® OS

Application Security User Guide for Security Devices

Published
2023-12-12

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

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.

Junos® OS Application Security User Guide for Security Devices

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

Use this guide to configure and operate Juniper Networks' AppSecure suite of application-aware security services in Junos OS on NFX Series and SRX Series Firewalls to provide visibility, enforcement, and control over the types of applications traversing in the networks.

RELATED DOCUMENTATION

JDPI-Decoder (Application Signature)

1

CHAPTER

Overview

Understanding Application Security

IN THIS SECTION

Web-based applications are changing the dynamics of security. Previously, specific applications were associated with specific protocols and ports, making policy enforcement at the host level relatively straightforward. Web applications that can be accessed from anywhere create challenge for network administrators to effectively manage traffic flows and access to data while delivering the security and network services.

An individual can connect to the network using multiple devices simultaneously, making it impractical to identify a user, an application, or a device by a group of statically allocated IP addresses and port numbers.

Applications such as instant messaging, peer-to-peer file sharing, Webmail, social networking, and IP voice/video collaboration evade security mechanisms by changing communications ports and protocols, or by tunneling within other commonly used services (for example, HTTP or HTTPS). Organizations need control over the applications and traffic on their networks to protect their assets against attacks and manage bandwidth.

Juniper Networks' AppSecure is a suite of application-aware security services for the Juniper Networks' SRX Series Firewalls and NFX Series devices to deliver security services to provide visibility and control over the types of applications traversing in the networks. AppSecure uses a sophisticated classification engine to accurately identify applications regardless of port or protocol, including nested applications that reside within trusted network services.

- Application identification (AppID)—Recognizes traffic at different network layers using characteristics other than port number. Once the application is determined, AppSecure service modules can be configured to monitor and control traffic for tracking, prioritization, access control, detection, and prevention based on the application ID of the traffic.
- Application Tracking (AppTrack)—Tracks and reports applications passing through the device.
- Application Firewall with Unified policies—Implements an application firewall functionality to block traffic based on specific dynamic applications using unified security policies.
- Application Quality of Service (AppQoS)—Provides quality-of-service prioritization based on application awareness.

- Advanced policy-based routing (APBR)— Classifies session based on applications and applies the configured rules to reroute the traffic.
- SSL Proxy—Provides visibility of encrypted traffic to allow deep packet inspection (DPI).

AppSecure works with additional content security through integrated Content Security , intrusion prevention systems (IPS), and Juniper Networks Juniper Advanced Threat Prevention Cloud (ATP Cloud) on the security devices for deeper protection against malware, spam, phishing, and application exploits.

Benefits of Application Security

- Helps you identify application traffic traversing your network regardless of port, protocol, and encryption, thereby providing greater visibility to control network traffic.
- Enables you to control network traffic by setting and enforcing security policies based on accurate application information.
- Provides context and clarity to strengthen network protection.
- Provides protection against common evasion techniques.

RELATED DOCUMENTATION

Application Identification | 5
Application Firewall | 130
Application Tracking | 166
Application QoS | 190
SSL Proxy | 350

2

CHAPTER

Application Identification

Application Identification

IN THIS SECTION

Application Identification enables you to see the applications on your network and learn how they work, their behavioral characteristics, and their relative risk. Using several different identification mechanisms, App ID detects the applications on your network regardless of the port, protocol, and other evasive tactics used. For more information, see the following topics:

Understanding Application Identification Techniques

IN THIS SECTION

Historically, firewalls have used the IP address and port numbers as a way of enforcing policies. That strategy is based on the assumption that users connect to the network from fixed locations and access particular resources using specific port numbers.

Today, wireless networking and mobile devices require a different strategy. The way in which devices connect to the network changes rapidly. An individual can connect to the network using multiple devices simultaneously. It is no longer practical to identify a user, application, or device by a group of statically allocated IP addresses and port numbers.

This topic includes the following section:

Junos OS Next-Generation Application Identification

Next-generation application identification builds on the legacy application identification functionality and provides more effective detection capabilities for evasive applications such as Skype, BitTorrent, and Tor.

Junos OS application identification recognizes Web-based and other applications and protocols at different network layers using characteristics other than port number. Applications are identified by using a protocol bundle containing application signatures and parsing information. The identification is based on protocol parsing and decoding and session management.

The detection mechanism has its own data feed and constructs to identify applications.

The following features are supported in application identification:

- Support for protocols and applications, including video streaming, peer-to-peer communication, social networking, and messaging
- Identification of services within applications
- Ability to distinguish actions launched within an application (such as login, browse, chat, and file transfer)
- Support for all versions of protocols and application decoders and dynamic updates of decoders
- Support for encrypted and compressed traffic and most complex tunneling protocols
- Ability to identify all protocols from Layer 3 to Layer 7 and above Layer 7

Benefits of Application Identification

- Provides granular control over applications, including video streaming, peer-to-peer communication, social networking, and messaging. It also identifies services, port usage, underlying technology, and behavioral characteristics within applications. This visibility enables you to block evasive applications inline at the SRX Series firewall.

- Identifies applications and allows, blocks, or limits applications— regardless of port or protocol, including applications known for using evasive techniques to avoid identification. This identification helps organizations control the types of traffic allowed to enter and exit the network.

Application Signature Mapping

Application signature mapping is a precise method of identifying the application that issued traffic on the network. Signature mapping operates at Layer 7 and inspects the actual content of the payload.

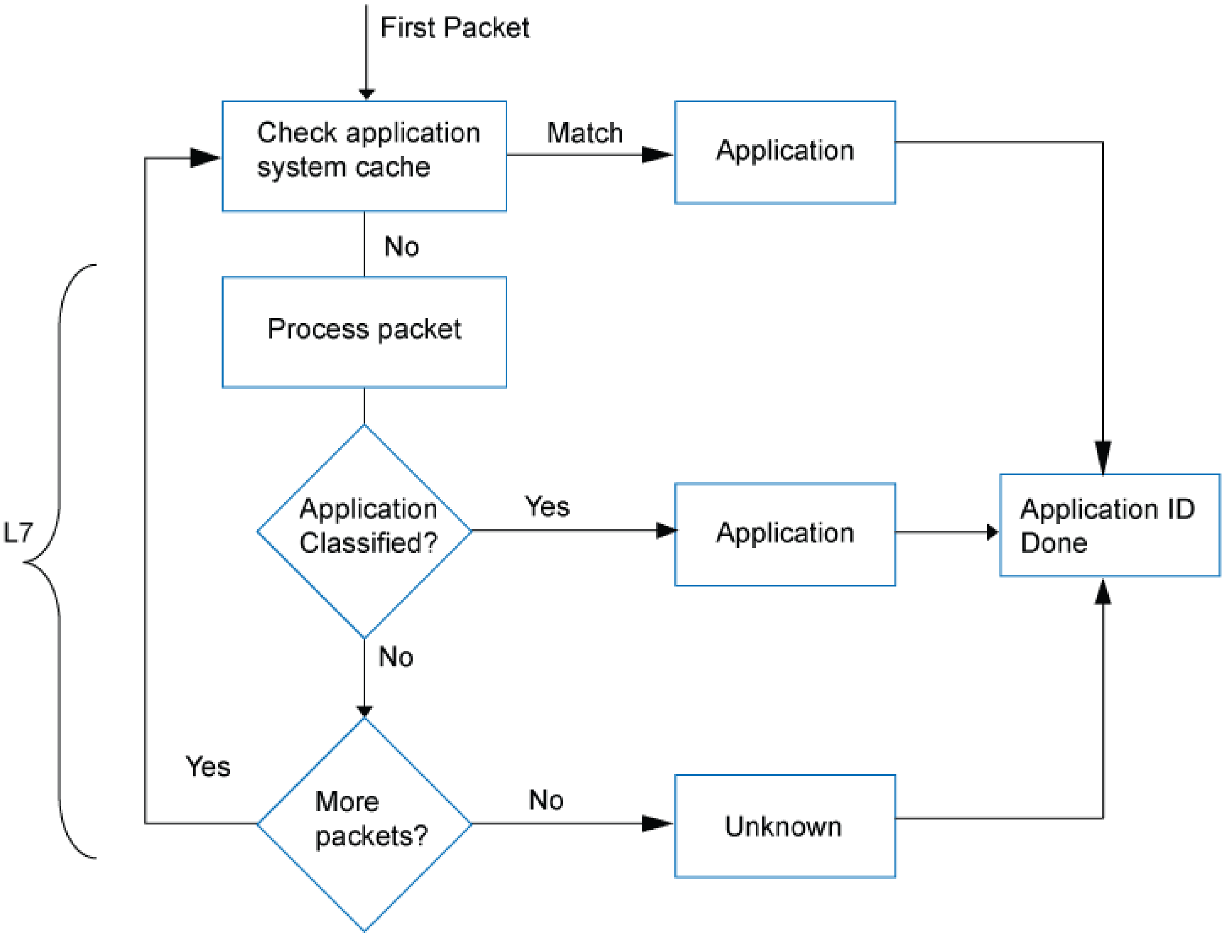
Applications are identified by using a downloadable protocol bundle. Application signatures and parsing information of the first few packets are compared to the content of the database. If the payload contains the same information as an entry in the database, the application of the traffic is identified as the application mapped to that database entry.

Juniper Networks provides a predefined application identification database that contains entries for a comprehensive set of known applications, such as FTP and DNS, and applications that operate over the HTTP protocol, such as Facebook, Kazaa, and many instant messaging programs. A signature subscription allows you to download the database from Juniper Networks and regularly update the content as new predefined signatures are added.

Application Identification Match Sequence

Figure 1 on page 8 shows the sequence in which mapping techniques are applied and how the application is determined.

Figure 1: Mapping Sequence



In application identification, every packet in the flow passes through the application identification engine for processing until the application is identified. Application bindings are saved in the application system cache (ASC) to expedite future identification process.

Application signatures identify an application based on protocol grammar analysis in the first few packets of a session. If the application identification engine has not yet identified the application, it passes the packets and waits for more data.

The application identification module matches applications for both client-to-server and server-to-client sessions.

Once the application is determined, AppSecure service modules can be configured to monitor and control traffic for tracking, prioritization, access control, detection, and prevention based on the application ID of the traffic.

- Application Tracking (AppTrack)— Tracks and reports applications passing through the device.

- Intrusion Detection and Prevention (IDP)— Applies appropriate attack objects to applications running on nonstandard ports. Application identification improves IDP performance by narrowing the scope of attack signatures for applications without decoders.
- Application Firewall (AppFW)— Implements an application firewall using application-based rules.
- Application Quality of Service (AppQoS)— Provides *quality-of-service* prioritization based on application awareness.
- Advanced policy-based routing (APBR)— Classifies session based on applications and applies the configured rules to reroute the traffic.
- Application Quality of Experience (AppQoE)— Monitors the performance of applications, and based on the score, selects the best possible link for that application traffic.

SEE ALSO

Understanding Application Tracking | **167**

Application Firewall Overview | **130**

Understanding Application Quality of Service (AppQoS) | **190**

Understanding the Junos OS Application Identification Database

A predefined signature database is available on the Juniper Networks Security Engineering website. This database includes a library of application signatures. See Application Signatures for more details. These signature pages will give you visibility into the application category, group, risk-level, ports, and so on.

The predefined signature package provides identification criteria for known application signatures and is updated periodically.

Whenever new applications are added, the protocol bundle is updated and generated for all relevant platforms. It is packaged together with other application signature files. This package will be available for download through the security download website.

A subscription service allows you to regularly download the latest signatures for up-to-date coverage without having to create entries for your own use.

Application identification is enabled by default and is automatically turned on when you configure Intrusion Detection and Prevention (IDP), AppFW, AppQoS, or AppTrack.

NOTE: Updates to the Junos OS predefined application signature package are authorized by a separately licensed subscription service. You must install the application identification application signature update license key on your device to download and install the signature database updates provided by Juniper Networks. When your license key expires, you can continue to use the locally stored application signature package contents but you cannot update the package.

SEE ALSO

Understanding the Junos OS Application Package Installation | 34

Understanding IDP Application Identification

Disabling and Reenabling Junos OS Application Identification

Application identification is enabled by default. You can disable application identification with the CLI.

To disable application identification:

```
user@host# set services application-identification no-application-identification
```

If you want to reenable application identification, delete the configuration statement that specifies disabling of application identification:

```
user@host# delete services application-identification no-application-identification
```

If you are finished configuring the device, commit the configuration.

To verify the configuration, enter the `show services application-identification` command.

SEE ALSO

Understanding Application Identification Techniques | 5

Understanding the Junos OS Application Identification Database | 9

Understanding the Application System Cache

Application system cache (ASC) saves the mapping between an application type and the corresponding destination IP address, destination port, protocol type, and service. Once an application is identified, its information is saved in the ASC so that only a matching entry is required to identify an application running on a particular system, thereby expediting the identification process.

By default, the ASC saves the mapping information for 3600 seconds. However, you can configure the cache timeout value by using the CLI.

You can use the `[edit services application-identification application-system-cache-timeout]` command to change the timeout value for the application system cache entries. The timeout value can be configured from 0 through 1,000,000 seconds. The ASC session might expire after 1000,000 seconds.

ASC entries expire after the configured ASC timeout. ASC entries are not refreshed even when there are cache hits (matching entry in ASC found) during the timeout period.

NOTE: When you configure a new custom application signature or modify an existing custom signature, all the existing application system cache entries for predefined and custom applications will be cleared.

NOTE: When you delete or disable a custom application signature, and the configuration commit fails, the application system cache (ASC) entry is not cleared completely; instead, a base application in the path of custom application will be reported in ASC.

SEE ALSO

Enabling or Disabling Application Groups in Junos OS Application Identification | 94

Enabling or Disabling Application System Cache for Application Services

Starting in Junos OS Release 18.2R1, the default behavior of the ASC is changed as follows:

- Before Junos OS Release 18.2R1—ASC is enabled by default for all services including security services.

- From Junos OS Release 18.2R1 onwards—ASC is enabled by default; note the difference in security services lookup:
    - ASC lookup for security services is not enabled by default. That is—security services including security policies, application firewall (AppFW), application tracking (AppTrack), application quality of service (AppQoS), Juniper ATP Cloud, IDP, and Content Security do not use the ASC by default.
    - ASC lookup for miscellaneous services is enabled by default. That is—miscellaneous services including advanced policy-based routing (APBR) use the ASC for application identification by default.

NOTE: The change in the default behavior of the ASC affects the legacy AppFW functionality. With the ASC disabled by default for the security services starting in Junos OS Release 18.2 onward, AppFW will not use the entries present in the ASC.

You can revert to the ASC behavior as in Junos OS releases before Release 18.2 by using the `set services application-identification application-system-cache security-services` command.

CAUTION: The security device might become susceptible to application evasion techniques if the ASC is enabled for security services. We recommend that you enable the ASC only when the performance of the device in its default configuration (disabled for security services) is not sufficient for your specific use case.

Use the following commands to enable or disable the ASC:

- Enable the ASC for security services:

```
user@host# set services application-identification application-system-cache security-services
```

- Disable the ASC for miscellaneous services:

```
user@host# set services application-identification application-system-cache no-miscellaneous-services
```

- Disable the enabled ASC for security services:

```
user@host# delete services application-identification application-system-cache security-services
```

- Enable the disabled ASC for miscellaneous services:

  ```
  user@host# delete services application-identification application-system-cache no-
  miscellaneous-services
  ```

You can use the `show services application-identification application-system-cache` command to verify the status of the ASC.

The following sample output provides the status of the ASC:

```
user@host>show services application-identification application-system-cache
Application System Cache Configurations:
  application-cache: on
    Cache lookup for security-services: off
    Cache lookup for miscellaneous-services: on
  cache-entry-timeout: 3600 seconds
```

In releases before Junos OS Release 18.2R1, application caching was enabled by default. You can manually disable it by using the `set services application-identification no-application-system-cache` command.

```
user@host# set services application-identification no-application-system-cache
```

SEE ALSO

Understanding Application Identification Techniques | 5

Verifying Application System Cache Statistics | 13

Understanding the Junos OS Application Identification Database | 9

Verifying Application System Cache Statistics

IN THIS SECTION

Purpose

Verify the application system cache (ASC) statistics.

NOTE: The application system cache will display the cache for application identification applications.

Action

From CLI operation mode, enter the `show services application-identification application-system-cache` command.

Sample Output

command-name

```
user@host> show services application-identification application-system-cache
application-cache: on
  nested-application-cache: on
  cache-unknown-result: on
  cache-entry-timeout: 3600 seconds
```

Meaning

The output shows a summary of the ASC statistics information. Verify the following information:

- IP address—Displays the destination address.
- Port—Displays the destination port on the server.
- Protocol—Displays the protocol type on the destination port.
- Application—Displays the name of the application identified on the destination port.

NOTE: On for SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices, when there are a large number of ASC entries (10,000 or more), and the entries are to be listed in the output for the command `show services application-identification application-system-cache`, a CLI session timeout occurs.

SEE ALSO

Understanding Application Identification Techniques | 5
Enabling or Disabling Application System Cache for Application Services | 11

Onbox Application Identification Statistics

IN THIS SECTION

Application Identification services provide statistical information per session. These statistics provide customers with an application usage profile. The Onbox Application Identification Statistics feature adds application-level statistics to the AppSecure suite. Application statistics allow an administrator to access cumulative statistics as well as statistics accumulated over user-defined intervals.

With this feature, the administrator can clear the statistics and configure the interval values while maintaining bytes and session count statistics. Because the statistics count occurs at session close event time, the byte and session counts are not updated until the session closes. Juniper Networks security devices support a history of eight intervals that an administrator can use to display application session and byte counts. Starting in Junos OS 18.3R1, the security devices support a history of one interval to display application session and byte counts.

If application grouping is supported in your configuration of Junos OS, then the Onbox Application Identification Statistic feature supports onbox per-group matching statistics. The statistics are maintained for predefined groups only.

Reinstalling an application signature package will not clear the application statistics. If the application is disabled, there will not be any traffic for that application, but the application is still maintained in the statistics. It does not matter if you are reinstalling a predefined application, because applications are tracked according to application type. For predefined group statistics, reinstalling a security package will not clear the statistics. However, any changes to group memberships are updated. For example, junos:web might have 50 applications in the current release and 60 applications following an upgrade. Applications that are deleted and application groups that are renamed are handled in the same way as applications that are added.

The Application Identification module maintains a 64-bit session counters for each application on each Services Processing Unit (SPU). The counter increments when a session is identified as a particular application. Another set of 64-bit counters aggregates the total bytes per application on the SPU. Counters for unspecified applications are also maintained. Statistics from multiple SPUs for both sessions and bytes are aggregated on the Routing Engine and presented to the users.

Individual SPUs have interval timers to roll over statistics per *interval* time. To configure the interval for statistics collection, use the `set services application-identification statistics interval` *`time`* command. Whenever the Routing Engine queries for the required interval, the corresponding statistics are fetched from each SPU, aggregated in the Routing Engine and presented to the user.

Use the `clear services application-identification statistics` to clear all application statistics such as cumulative, interval, applications, and application groups.

Use the `clear services application-identification counter` command to reset the counters manually. Counters reset automatically when a device is upgraded or rebooted, when flowd restarts, or when there is a change in the interval timer.

Use the `set services application-identification application-system-cache-timeout value` to specify the timeout value in seconds for the application system cache entries.

Starting from Junos OS Release 15.1X49-D120, on all SRX Series Firewalls, the default time interval for application identification statistics collection time is changed from 1 minute to 1440 minutes.

Configuring IMAP Cache Size

Internet Message Access Protocol (IMAP) is an Internet standard protocol used by e-mail clients for e-mail storage and retrieval services. IMAP cache is used for protocol parsing and context generation. It stores parsing related information of an email.

Starting from Junos OS Release 15.1X49-D120, you can configure to limit the maximum number of entries in the IMAP cache and specify the timeout value for the entries in the cache.

You can use the following commands to modify the settings for IMAP cache:

```
set services application-identification imap-cache imap-cache-size size
```

```
set services application-identification imap-cache imap-cache-timeout time in seconds
```

Example:

```
[edit]
user@host# set services application-identification imap-cache imap-cache-size 50000
```

In this example, the IMAP cache size is configured to store 50,000 entries.

```
[edit]
user@host# set services application-identification imap-cache-timeout 600
```

In this example, time out period is configured to 600 seconds during which a cache entry remains in IMAP cache.

SEE ALSO

> Understanding Application Identification Techniques | 5

Understanding Jumbo Frames Support for Junos OS Application Identification Services

Application identification support the larger jumbo frame size of 9192 bytes. Although jumbo frames are enabled by default, you can adjust the maximum transmission unit (MTU) size by using the [`set interfaces`] command. CPU overhead can be reduced while processing jumbo frames.

SEE ALSO

> Understanding Jumbo Frames Support for Ethernet Interfaces

Application Identification Inspection Limit

IN THIS SECTION

Starting in Junos OS Releases 15.1X49-D200 and 19.4R1, you have the flexibility to configure the application identification inspection limits:.

- **Inspection Limit for TCP and UDP Sessions**

  You can set the byte limit and the packet limit for application identification (AppID) in a UDP or in a TCP session. AppID concludes the classification based on the configured inspection limit. On exceeding the limit, AppID terminates the application classification.

  If AppID does not conclude the final classification within the configured limits, and a pre-matched application is available, AppID concludes the application as the pre-matched application. Otherwise, the application is concluded as junos:UNKNOWN provided the global AppID cache is enabled. The global AppID cache is enabled by default.

  To configure the byte limit and the packet limit, use the following configuration statements from the `[edit]` hierarchy:

  -
    ```
    user@host# set services application-identification inspection-limit tcp byte-limit byte-limit-number packet-limit packet-limit-number
    ```

  -
    ```
    user@host# set services application-identification inspection-limit udp byte-limit byte-limit-number packet-limit packet-limit-number
    ```

  Table 1 on page 19 provides the range and default value for configuring the byte limit and the packet limit for TCP and UDP sessions.

Table 1: Maximum Byte Limit and Packet Byte Limit for TCP and UDP Sessions

Session	Limit	Range	Default Value
TCP	Byte limit	0 through 4294967295	6000 For Junos OS Release 15.1X49-D200, the default value is 10000.
	Packet limit	0 through 4294967295	Zero
UDP	Byte limit	0 through 4294967295	Zero
	Packet limit	0 through 4294967295	10 For Junos OS Release 15.1X49-D200, the default value is 20.

The byte limit excludes the IP header and the TCP/UDP header lengths.

If you set the both the `byte-limit` and the `packet-limit` options, AppID inspects the session until both the limits are reached.

You can disable the TCP or UDP inspection limit by configuring the corresponding `byte-limit` and the `packet-limit` values to zero.

- **Global Offload Byte Limit (Other Sessions)**

  You can set the byte limit for the AppID to conclude the classification and identify the application in a session. On exceeding the limit, AppID terminates the application classification and takes one of the following decisions:

  - If a pre-matched application is available, AppID concludes the application classification as the pre-matched application in following cases:

    - When AppID does not conclude the final classification within the configured byte limit

    - When the session is not offloaded due to tunnelling behavior of some applications

  - If a pre-matched application is not available, AppID concludes the application as junos:UNKNOWN, provided the global AppID cache is enabled. The global AppID cache is enabled by default. See "Enabling or Disabling Application System Cache for Application Services" on page 11.

To configure the byte limit, use the following configuration statement from the [edit] hierarchy:

```
set services application-identification global-offload-byte-limit  byte-limit-number
```

The default value for the global-offload-byte-limit option is 10000.

You can disable the global offload byte limit by configuring the global-offload-byte-limit value to zero.

The byte limit excludes the IP header and the TCP/UDP header lengths.

Enable Performance Mode Option

Starting in Junos OS Releases 15.1X49-D200 and 19.4R1, the maximum packet threshold for DPI performance mode option set services application-identification enable-performance-mode max-packet-threshold *value* is deprecated—rather than immediately removed—to provide backward compatibility and an opportunity to bring your configuration into compliance with the new configuration. This option was used for setting the maximum packet threshold for the DPI performance mode.

If your configuration includes enabled performance mode option with max-packet-threshold in Junos OS 15.1X49-D200 and 19.4R1 releases, AppID concludes the application classification on reaching the lowest value configured in the TCP or UDP inspection limit or global offload byte limit, or in the maximum packet threshold for DPI performance mode option.

Application Identification Support for Applications Hosted on Content Delivery Network (CDN)

Starting in Junos OS Release 20.1R1 and 19.1R3, you can enable application identification (AppID) to classify a web application that is hosted on a content delivery network (CDN) such as AWS, Akamai, Azure, Fastly, and Cloudflare and so on accurately. Use the following configuration statement to enable CDN application classification:

```
[edit]
user@host# user@hots# set service application-identification enable-cdn-application-detection
```

When you apply the configuration, AppID identifies and classifies actual applications that are hosted on the CDN.

Maximum Memory Limit for DPI

Starting in Junos OS Release 20.1R1 and 19.1R3, you can configure the maximum memory limit for deep packet inspection (DPI) by using the following configuration statement:

```
user@host# set services application-identification max-memory memory-value
```

You can set 1 through 200000 MB as memory value.

Once the JDPI memory consumption reaches to 90% of the configured value, then DPI stops processing new sessions.

Improving the Application Traffic Throughput

The application traffic throughput can be improved by setting the deep packet inspection (DPI) in performance mode with default packet inspection limit as two packets, including both client-to-server and server-to-client directions. By default, performance mode is disabled on security devices.

To improve the application traffic throughput:

1. Enable the DPI performance mode.

   ```
   [edit]
   user@host# set services application-identification enable-performance-mode
   ```

2. (Optional) You can set the maximum packet threshold for DPI performance mode, including both client-to-server and server-to-client directions.

   You can set the packet inspection limit from 1 through 100.

   ```
   [edit]
   user@host# set services application-identification enable-performance-mode max-packet-
   threshold value
   ```

   Starting in Junos OS Releases 15.1X49-D200 and 19.4R1, the maximum packet threshold for DPI performance mode option `set services application-identification enable-performance-mode max-packet-threshold` *value* is deprecated—rather than immediately removed—to provide backward compatibility and an opportunity to bring your configuration into compliance with the new configuration. This option was used for setting the maximum packet threshold for the DPI performance mode.

3. Commit the configuration.

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

Use the `show services application-identification status` command to display detailed information about application identification status.

show services application-identification status (DPI Performance Mode Enabled)

```
user@host> show services application-identification status
pic: 2/1

Application Identification
Status                            Enabled
Sessions under app detection      0
Engine Version                    4.18.2-24.006 (build date Jul 30 2014)
Max TCP session packet memory     30000
Force packet plugin               Disabled
Force stream plugin               Disabled
DPI Performance mode:             Enabled
Statistics collection interval    1 (in minutes)

Application System Cache
Status                            Enabled
Negative cache status             Disabled
Max Number of entries in cache    262144
Cache timeout                     3600 (in seconds)

Protocol Bundle
Download Server                   https://signatures.juniper.net/cgi-bin/index.cgi
AutoUpdate                        Disabled
Slot 1:
Application package version       2399
Status                            Active
Version                           1.40.0-26.006 (build date May 1 2014)
Sessions                          0
Slot 2
Application package version       0
Status                            Free
```

```
Version
Sessions                          0
```

The DPI Performance mode field displays whether the DPI performance mode is enabled or not. This field is displayed in the CLI command output only if the performance mode is enabled.

If you want to set DPI to default accuracy mode and disable the performance mode, delete the configuration statement that specifies enabling of the performance mode:

To disable the performance mode:

1. Delete the performance mode.

   ```
   [edit]
   user@host# delete services application-identification enable-performance-mode
   ```

2. Commit the configuration.

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

SEE ALSO

> *enable-performance-mode*

Packet Capture of Unknown Application Traffic Overview

IN THIS SECTION

You can use the packet capture of unknown applications feature to gather more details about an unknown application on your security device. Unknown application traffic is the traffic that does not match an application signature.

Once you've configured packet capture options on your security device, the unknown application traffic is gathered and stored on the device in a packet capture file (**.pcap**). You can use the packet capture of an unknown application to define a new custom application signature. You can use this custom application signature in a security policy to manage the application traffic more efficiently.

You can send the **.pcap** file to Juniper Networks for analysis in cases where the traffic is incorrectly classified, or to request creation of an application signature.

Benefits of Packet Capture of Unknown Application Traffic

You can use the packet capture of unknown application traffic to:

- Gather more insight about an unknown application
- Analyze unknown application traffic for potential threats
- Assist in creation of security policy rules
- Enable custom application signature creation

NOTE: Implementing security policies that block all unknown application traffic could cause issues with network-based applications. Before applying these types of policies, be sure to validate that this approach does not cause issues in your environment. You must carefully analyze the unknown application traffic, and define the security policy accordingly.

Configure Packet Capture For Unknown Application Traffic

IN THIS SECTION

Before You Begin

To enable automatic packet capture of unknown application traffic, you must:

- Install a valid application identification feature license on your SRX Series Firewall. See Managing Junos OS Licenses.
- Download and install the Junos OS application signature package. See Download and Install Junos OS Application Signature Package.
- Ensure you have Junos OS Release 20.2R1 or later version on your security device.

Overview

In this example, you'll learn how to configure automated packet capture of unknown applications on your security device by completing the following steps:

- Set packet capture options at global level or at a security policy level.
- Configure packet capture mode
- (Optional) Configure packet capture file options
- Access the generated packet capture file (**.pcap** file)

Configuration

IN THIS SECTION

To learn about packet capture configuration options, see packet-capture before you begin.

Packet Capture for Unknown Applications Globally

Step-by-Step Procedure

- To enable packet capture at a global level, use the following command:

```
user@host# set services application-identification packet-capture global
```

When you enable packet capture at the global level, your security device generates a packet capture for all sessions that contain unknown application traffic.

Packet Capture for Unknown Applications At a Security Policy Level

Step-by-Step Procedure

- Configure packet capture at a security policy level, use the following procedure. In this example, you'll enable packet capture of unknown application traffic at the security policy P1.

```
[edit]
user@host# set security policies from-zone untrust to-zone trust policy P1 match source-
address any
user@host# set security policies from-zone untrust to-zone trust policy P1 match destination-
address any
user@host# set security policies from-zone untrust to-zone trust policy P1 match application
any
user@host# set security policies from-zone untrust to-zone trust policy P1 match dynamic-
application junos:UNKNOWN
user@host# set security policies from-zone untrust to-zone trust policy P1 then permit
application-services packet-capture
```

To enable packet capture of unknown application traffic at the security policy level, you must include `junos:UNKNOWN` as the dynamic-application match conditions.

When you configure the security policy (P1), the system captures the packet details for the application traffic that matches the security policy match criteria.

Selecting Packet Capture Mode

You can capture the packets for the unknown application traffic in either of the following modes:

- ASC mode—Captures packets for unknown applications when the application is classified as junos:UNKNOWN and has a matching entry in the application system cache (ASC). This mode is enabled by default.

- Aggressive mode—Captures all traffic before AppID has finished classification. In this mode, the system captures all application traffic regardless of an available ASC entry. Packet capture begins from the first packet of the first session. Note that aggressive mode is significantly more resource-intensive and should be used with caution.

  To enable aggressive mode, use the following command:

  ```
  [edit]
  user@host# set services application-identification packet-capture aggressive-mode
  ```

  We do not recommend using aggressive mode unless you need to capture the first occurrence of a flow. As noted above, the default behavior of the device relies on the ASC.

Define Packet Capture Options (Optional)

Step-by-Step Procedure

Optionally, you can set the following packet capture parameters. Otherwise, the default options described in packet-capture are used for this feature. In this example, you define packet capture options such as maximum packet limit, maximum byte limit, and number of packet capture (.pcap) files.

1. Set the maximum number of UDP packets per session.

   ```
   [edit]
   user@host# set services application-identification packet-capture max-packets 10
   ```

2. Set the maximum number of TCP bytes per session.

   ```
   [edit]
   user@host# set services application-identification packet-capture max-bytes 2048
   ```

3. Set the maximum number of packet capture (.pcap) files to be created before the oldest one is overwritten and rotated out.

   ```
   [edit]
   user@host# set services application-identification packet-capture max-files 30
   ```

Results

From configuration mode, confirm your configuration by entering the `show services application-identification packet-capture` command and `show security policies` hierarchy level. If the output does not display the intended configuration, follow the configuration instructions in this example to correct it.

The following configuration shows an example of unknown application packet capture at the global level with optional configurations:

```
[edit services application-identification]
user@host# show packet-capture
{
    global;
    max-packets 10;
    max-bytes 2048;
    max-files 30;
}
```

The following configuration shows an example of unknown application packet capture at a security policy level with optional configurations:

```
[edit services application-identification]
user@host# show packet-capture
{
    max-packets 10;
    max-bytes 2048;
    max-files 30;
}
```

```
[edit security policies]
user@host# show
from-zone untrust to-zone trust {
    policy P1 {
```

```
        match {
            source-address any;
            destination-address any;
            application any;
            dynamic-application [ junos:UNKNOWN ];
        }
        then {
            permit {
                application-services {
                    packet-capture;
                }
            }
        }
    }
}
```

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

Accessing Packet Capture Files (.pcaps)

After you complete the configuration and commit it, you can view the packet capture (**.pcap**) file. The system generates a unique packet capture file for each destination IP address, destination port, and protocol.

Step-by-Step Procedure

To view the packet capture file:

1. Navigate to the directory where **.pcap** files are stored on the device.

   ```
   user@host> start shell
   %
    % cd /var/log/pcap
   ```

2. Locate the **.pcap** file.

   The **.pcap** file is saved in *destination-IP-address. destination-port.protocol. pcap* format. Example: **142.250.31.156_443_17.pcap**.

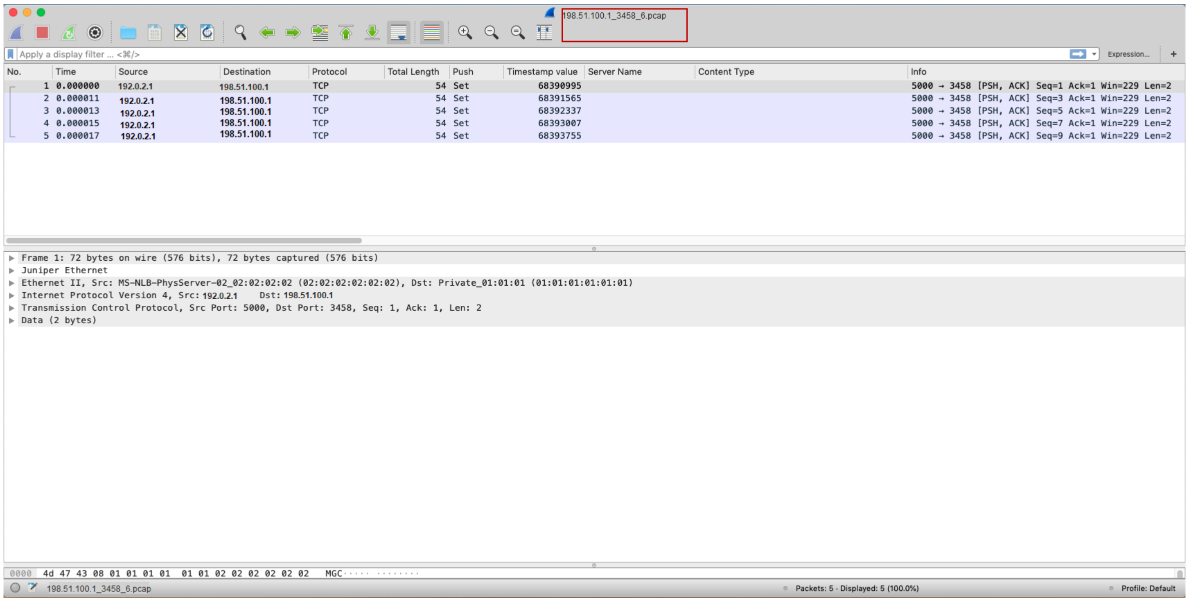
   ```
   user@host:/var/log/pcap # ls -lah
   total 1544
   drwxr-xr-x  2 root  wheel   3.0K Jul 27 15:04 .
   ```

```
drwxrwxr-x  9 root  wheel   3.0K Jul 24 16:23 ..
-rw-r-----  1 root  wheel   5.0K Jul 24 20:16 142.250.31.156_443_17.pcap
-rw-r-----  1 root  wheel    16K Jul 27 15:03 142.250.64.97_443_17.pcap
-rw-r-----  1 root  wheel   9.0K Jul 27 14:26 162.223.228.170_443_17.pcap
-rw-r-----  1 root  wheel   2.1K Jul 26 17:06 17.133.234.32_16385_17.pcap
-rw-r-----  1 root  wheel    11K Jul 24 16:20 172.217.0.226_443_17.pcap
-rw-r-----  1 root  wheel    16K Jul 27 14:21 172.217.9.234_443_17.pcap
-rw-r-----  1 root  wheel    31K Jul 27 14:25 172.217.9.238_443_17.pcap
-rw-r-----  1 root  wheel    17K Jul 24 19:21 52.114.132.87_3478_17.pcap
```

You can download the **.pcap** file by using SFTP or SCP and view it with Wireshark or your favorite network analyzer.

Figure 2 on page 30 shows a sample **.pcap** file generated for the unknown application traffic.

Figure 2: Sample Packet Capture File



No.	Time	Source	Destination	Protocol	Total Length	Push	Timestamp value	Server Name	Content Type	Info
1	0.000000	192.0.2.1	198.51.100.1	TCP	54	Set	68390995			5000 → 3458 [PSH, ACK] Seq=1 Ack=1 Win=229 Len=2
2	0.000011	192.0.2.1	198.51.100.1	TCP	54	Set	68391565			5000 → 3458 [PSH, ACK] Seq=3 Ack=1 Win=229 Len=2
3	0.000013	192.0.2.1	198.51.100.1	TCP	54	Set	68392337			5000 → 3458 [PSH, ACK] Seq=5 Ack=1 Win=229 Len=2
4	0.000015	192.0.2.1	198.51.100.1	TCP	54	Set	68393007			5000 → 3458 [PSH, ACK] Seq=7 Ack=1 Win=229 Len=2
5	0.000017	192.0.2.1	198.51.100.1	TCP	54	Set	68393755			5000 → 3458 [PSH, ACK] Seq=9 Ack=1 Win=229 Len=2

```
▶ Frame 1: 72 bytes on wire (576 bits), 72 bytes captured (576 bits)
▶ Juniper Ethernet
▶ Ethernet II, Src: MS-NLB-PhysServer-02_02:02:02:02 (02:02:02:02:02:02), Dst: Private_01:01:01 (01:01:01:01:01:01)
▶ Internet Protocol Version 4, Src: 192.0.2.1     Dst: 198.51.100.1
▶ Transmission Control Protocol, Src Port: 5000, Dst Port: 3458, Seq: 1, Ack: 1, Len: 2
▶ Data (2 bytes)
```

```
0000  4d 47 43 08 01 01 01 01  01 01 02 02 02 02 02 02   MGC····· ········
```

198.51.100.1_3458_6.pcap   Packets: 5 · Displayed: 5 (100.0%)   Profile: Default

NOTE: In situations where packet loss is occurring, the device may not be able to capture all relevant details of the flow. In this case, the **.pcap** file will only reflect what the device was able to ingest and process.

The security device saves the packet capture details for all traffic that matches the three match criteria (destination IP address, destination port, and protocol) in the same file regardless of global or policy-level configuration. The system maintains the cache with the destination IP address, destination port, and the protocol and does not accept the repeated capturing of the same traffic which exceeds the defined limit. You can set the packet capture file options as in packet-capture.

Verification

IN THIS SECTION

Viewing Packet Capture Details

Purpose

View the packet capture details to confirm that your configuration is working.

Action

Use the `show services application-identification packet-capture counters` command.

```
user@host> show services application-identification packet-capture counters

pic: 0/0
  Counter type                                                      Value
Total sessions captured                                                 47
Total packets captured                                                  282
Active sessions being captured                                          1
Sessions ignored because of memory allocation failures                  0
Packets ignored because of memory allocation failures                   0
Ipc messages ignored because of storage limit                           0
Sessions ignored because of buffer-packets limit                        0
Packets ignored because of buffer-packets limit                         0
Inconclusive sessions captured                                          4
Inconclusive sessions ignored                                           0
Cache entries timed out                                                 0
```

Meaning

From this sample output, you can get details such as the number of sessions being captured, and the number of sessions already captured. For more details about the packet capture counters, see show services application-identification packet-capture counters.

Packet Capture of Unknown Applications Details per Session

Starting in Junos OS Release 21.1, your security device stores the packet capture of unknown applications details per session. As a result of this change, the packet capture (.pcap) file now includes the session ID in the file name. That is—destination-IP-address_destination-port_protocol_session-id.pcap in /var/log/pcap location.

By storing the packet capture per session, the .pcap file size is reduced as it saves details per session only.

In addition, we've enhanced packet capture of unknown application functionality to capture unknown SNI details

SEE ALSO

- request services application-identification clear packet-capture all
- clear services application-identification packet-capture counters

Release History Table

Release	Description
19.4R1	Starting in Junos OS Releases 15.1X49-D200 and 19.4R1, you have the flexibility to configure the application identification inspection limits:
19.4R1	Starting in Junos OS Releases 15.1X49-D200 and 19.4R1, the maximum packet threshold for DPI performance mode option `set services application-identification enable-performance-mode max-packet-threshold` *`value`* is deprecated
18.2R1	Starting in Junos OS Release 18.2R1, the default behavior of the ASC is changed
18.2R1	In releases before Junos OS Release 18.2R1, application caching was enabled by default. You can manually disable it by using the `set services application-identification no-application-system-cache` command.

15.1X49-D120	Starting from Junos OS Release 15.1X49-D120, you can configure to limit the maximum number of entries in the IMAP cache and specify the timeout value for the entries in the cache.

RELATED DOCUMENTATION

Understanding Application Security | 2

Predefined Application Signatures for Application Identification | 33

Custom Application Signatures for Application Identification | 72

Predefined and Custom Application Groups for Application Identification | 87

Predefined Application Signatures for Application Identification

IN THIS SECTION

Predefined application signature package is a dynamically loadable module that provides application classification functionality and associated protocol attributes. It is hosted on an external server and can be downloaded as a package and installed on the device. For more information, see the following topics:

Understanding the Junos OS Application Package Installation

IN THIS SECTION

Juniper Networks regularly updates the predefined application signature package database and makes it available to subscribers on the Juniper Networks website. This package includes signature definitions of known application objects that can be used to identify applications for tracking, firewall policies, *quality-of-service* prioritization, and Intrusion Detection and Prevention (IDP). The database contains application objects such as FTP, DNS, Facebook, Kazaa, and many instant messenger programs.

You need to download and install the application signature package before configuring application services. The application signature package is included in the IDP installation directly and does not need to be downloaded separately.

- If you have IDP enabled and plan to use application identification, you can continue to run the IDP signature database download. To download the IDP signature database, run the following command: `request security idp security-package download`. The application package download can be performed manually or automatically. See "Downloading and Installing the Junos OS Application Signature Package As Part of the IDP Security Package" on page 44.

  NOTE: If you have an IDP-enabled device and plan to use application identification, we recommend that you download only the IDP signature database. This will avoid having two versions of the application database, which could become out of sync.

- If you do not have IDP enabled and plan to use application identification, you can run the following commands: `request services application-identification download` and `request services application-identification install`. These commands will download the application signature database and install it on the device.

  You can perform the download manually or automatically. When you download the extracted package manually, you can change the download URL.

After downloading and installing the application signature package, use CLI commands to download and install database updates, and view summary and detailed application information.

See "Downloading and Installing the Junos OS Application Signature Package Manually" on page 39 or "Example: Scheduling the Application Signature Package Updates" on page 53.

NOTE: The Junos OS application signature package update is a separately licensed subscription service. You must install the application signature package update license key on your device to download and install the signature database updates provided by Juniper Networks. If your license key expires, you can continue to use the locally stored application signature package content but you cannot update the data.

NOTE: Starting from Junos OS Release 15.1X49-D50 and Junos OS Release 17.3, when you upgrade or downgrade an application signature package, an error message is displayed if there is any mismatch of application IDs (unique ID number of an application signature) between proto bundles and these applications are configured in AppFW and AppQoS rules. Example:

```
Please resolve following references and try it again
[edit class-of-service application-traffic-control rule-sets RS8 rule 1 match application
junos:CCPROXY]
```

As a workaround, disable the AppFW and AppQoS rules before upgrading or downgrading an application signature package. You can reenable AppFW and AppQoS rules once the upgrade or downgrade procedure is complete.

NOTE: On all security devices, J-Web pages for AppSecure Services are preliminary. We recommend using the CLI for configuration of AppSecure features.

NOTE: This feature requires a license. To understand more about Junos OS application signature package, Please refer to the Juniper Licensing Guide for general information about License Management. Please refer to the product Data Sheets at SRX Series Services Gateways for details, or contact your Juniper Account Team or Juniper Partner.

Upgrading to Next-Generation Application Identification

Starting from Junos OS Release 12.1X47-D10, next-generation application identification is supported. You must install Junos OS Release 12.1X47-D10 to migrate from existing, or legacy, application identification to next-generation application identification.

Security devices installed with Junos OS builds with legacy application identification include legacy application identification security packages. When you upgrade these devices with Junos OS Release 12.1X47-D10, the next-generation application identification security package is installed along with the default protocol bundle. The device is automatically upgraded to next-generation application identification.

NOTE:

- The next-generation application identification security package introduces incremental updates to the legacy application identification package. You are not required to remove or uninstall any existing applications.

- Applications supported in previous releases (Junos OS Release 12.1X46 or prior) might have new aliases or alternative names in the new version. So existing configurations using such application work in Junos OS Release 12.1X47; however, related logs and other information will use the new name. You can use the `show services application-identification application detail` *`new-application-name`* command to get the details of the applications.

- When you upgrade Junos OS, you can include the `validate` or `no-validate` options with the `request system software add` command. Because the existing features, which are not part of next-generation application identification, are deprecated, incompatibility issues are not seen.

- Next-generation application identification eliminates the generation of new nested applications and treats existing nested applications as normal applications. In addition, next-generation application identification does not support custom applications or custom application groups. Existing configurations involving any nested applications, custom applications, or custom application groups are ignored with warning messages.

SEE ALSO

Understanding the Junos OS Application Identification Database | 9

Understanding the IDP Signature Database

Downloading and Installing the Junos OS Application Signature Package Manually | 39

Downloading and Installing the Junos OS Application Signature Package As Part of the IDP Security Package | 44

Installing and Verifying Licenses for an Application Signature Package

The Junos OS application signature package update is a separately licensed subscription service. You must install the application signature package update license key on your device to download and install the signature database updates provided by Juniper Networks. If your license key expires, you can continue to use the locally stored application signature package content.

Licensing is usually ordered when the device is purchased, and this information is bound to the chassis serial number. These instructions assume that you already have the license. If you did not order the license during the purchase of the device, contact your account team or Juniper customer care for assistance. For more information, refer to the Knowledge Base article KB9731 at https://kb.juniper.net/InfoCenter/index?page=home.

Starting from Junos OS 15.1X49-D30 and Junos OS Release 17.3R1, on SRX1500 devices, AppSecure is part of Junos Software Enhanced (JSE) software license package. There is no separate license key for AppSecure is available. You must use JSE software license on your device to download and install the AppID signature database updates, or to use other AppSecure features such as AppFW, AppQoS, and AppTrack.

Starting from Junos OS 15.1X49-D30 and Junos OS Release 17.3R1, on SRX300, SRX320, SRX340, and SRX345 devices, AppSecure is part of Junos Software Enhanced (JSE) software license package. There is no separate license key for AppSecure is available. You must use JSE software license on your device to download and install the AppID signature database updates, or to use other AppSecure features such as AppFW, AppQoS, and AppTrack.

Starting from 15.1X49-D65 and Junos OS Release 17.3R1, on SRX4100, and SRX4200 devices, AppSecure is part of Junos Software Enhanced (JSE) license package. There is no separate license key for AppSecure is available. You must use JSE software license on your device to download and install the AppID signature database updates, or to use other AppSecure features such as AppFW, AppQoS, and AppTrack.

Junos Software Base (JSB) package does not include application signatures. Please refer to the product Data Sheets at SRX Series Services Gateways for details, or contact your Juniper Account Team or Juniper Partner.

You can install the license on the SRX Series Firewall using either the automatic method or manual method as follows:

- Install your license automatically on the device.

To install or update your license automatically, your device must be connected to the Internet .

```
user@host> request system license update
```

```
Trying to update license keys from https://ae1.juniper.net, use 'show system license' to
check status.
```

- Install the licenses manually on the device.

```
user@host> request system license add terminal
```

```
[Type ^D at a new line to end input,
 enter blank line between each license key]
```

Paste the license key and press Enter to continue.

- Verify the license is installed on your device.

Use the **show system license** `command` command to view license usage, as shown in the following example:

```
License usage:
                                 Licenses     Licenses    Licenses    Expiry
  Feature name                       used    installed      needed
  logical-system                        4            1           3    permanent

 License identifier: JUNOSXXXXXX
  License version: 2
  Valid for device: AA4XXXX005
  Features:
    appid-sig        - APPID Signature
      date-based, 2014-02-17 08:00:00 GMT-8 - 2015-02-11 08:00:00 GMT-8
```

The output sample is truncated to display only license usage details.

SEE ALSO

Adding New Licenses (CLI Procedure)

Downloading and Installing the Junos OS Application Signature Package Manually

IN THIS SECTION

This example shows how to download the application signature package, create a policy, and identify it as the active policy.

Requirements

Before you begin:

- Ensure that your security device has a connection to the Internet to download security package updates.

  NOTE: DNS must be set up because you need to resolve the name of the update server.

- Ensure that you have installed the application identification feature license.

This example uses the following hardware and software components:

- An SRX Series device
- Junos OS Release 12.1X47-D10

Overview

Juniper Networks regularly updates the predefined application signature package database and makes it available on the Juniper Networks website. This package includes application objects that can be used in Intrusion Detection and Prevention (IDP), application firewall policy, and AppTrack to match traffic.

Configuration

IN THIS SECTION

CLI Quick Configuration

CLI quick configuration is not available for this example because manual intervention is required during the configuration.

Downloading and Installing Application Identification

Step-by-Step Procedure

1. Download the application package.

   ```
   user@host> request services application-identification download
   ```

   ```
   Please use command "request services application-identification download status" to check
   status
   ```

   Download retrieves the application package from the Juniper Networks security website https://signatures.juniper.net/cgi-bin/index.cgi.

   You can also download a specific version of the application package or download the application package from the specific location by using the following options:

- To download a specific version of the application package:

  ```
  user@host>request services application-identification download version version-number
  ```

- To change the download URL for the application package from configuration mode:

  ```
  [edit]
  user@host# set services application-identification download url URL or File Path
  ```

  NOTE: If you change the download URL and you want to keep that change, make sure you commit the configuration.

2. Check the download status.

   ```
   user@host>request services application-identification download status
   ```

   ```
   Application package 2345 is downloaded successfully
   ```

   NOTE: You can also use the system log to view the result of the download. Starting in Junos OS Release 20.4R1, system log messages are updated to display the application signature package download and installation results.

3. Install the application package.

   ```
   user@host>request services application-identification install
   ```

   ```
   Please use command "request services application-identification install status" to check
   status and use command "request services application-identification proto-bundle-status" to
   check protocol bundle status
   ```

   The application package is installed in the application signature database on the device.

4. Check the installation status of the application package.

The command output displays information about the downloaded and installed versions of the application package and protocol bundle.

- To view the installation status:

```
user@host>request services application-identification install status
```

```
Install application package 2345 succeed
```

- To view the protocol bundle status:

```
user@host>request services application-identification proto-bundle-status
```

```
Protocol Bundle Version (1.30.4-22.005 (build date Jan 17 2014)) and application secpack
version (2345) is loaded and activated.
```

NOTE: It is possible that an application signature was removed from the newer version of an application signature database. If this signature is used in an existing application firewall policy on your device, the installation of the new database will fail. An installation status message identifies the signature that is no longer valid. To update the database successfully, remove all references to the deleted signature from your existing policies and groups, and rerun the install command.

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying the Application Identification Status

Purpose

Verify that the application identification configuration is working properly.

Action

From operational mode, enter the `show services application-identification status` command.

```
pic: 1/0

Application Identification
 Status                            Enabled
 Sessions under app detection      0
 Engine Version                    4.18.1-20 (build date Jan 25 2014)
 Max TCP session packet memory     30000
 Max C2S bytes                     1024
 Max S2C bytes                     0
 Force packet plugin               Disabled
 Force stream plugin               Disabled
 Statistics collection interval    1 (in minutes)

Application System Cache
 Status                            Enabled
 Negative cache status             Disabled
 Max Number of entries in cache    131072
 Cache timeout in seconds          3600

Protocol Bundle
 Download Server                   https://signatures.juniper.net/cgi-bin/index.cgi
 AutoUpdate                        Enabled
Slot 1:
 Status                            Active
 Version                           1.30.4-22.005 (build date Jan 17 2014)
 Sessions                          0
Slot 2
 Status                            Free
```

Meaning

The `Status: Enabled` field shows that application identification is enabled on the device.

SEE ALSO

Understanding the Junos OS Application Package Installation | 34
Example: Scheduling the Application Signature Package Updates | 53
Verifying the Junos OS Application Identification Extracted Application Package | 63
Uninstalling the Junos OS Application Identification Application Package | 65

Downloading and Installing the Junos OS Application Signature Package As Part of the IDP Security Package

IN THIS SECTION

You can download and install application signatures through intrusion detection and prevention (IDP) security packages.

This example shows how to enhance security by downloading and installing the IDP signatures and application signature package. In this case, both IDP signature pack and application signature pack are downloaded with a single command.

Requirements

Before you begin:

- Ensure that your SRX Series Firewall has a connection to the Internet to download security package updates.

NOTE: DNS must be set up because you need to resolve the name of the update server.

- Ensure that you have installed the application identification feature license.

This example uses the following hardware and software components:

- An SRX Series Firewall
- Junos OS Release 12.1X47-D10

Overview

In this example, you download and install the signature database from the Juniper Networks website.

Configuration

IN THIS SECTION

Downloading and Installing the Signature Database

CLI Quick Configuration

CLI quick configuration is not available for this example because manual intervention is required during the configuration.

Step-by-Step Procedure

To download and install application signatures:

1. Download the signature database.

```
[edit]
user@host# run request security idp security-package download
```

```
Will be processed in async mode. Check the status using the status checking CLI
```

NOTE: Downloading the database might take some time depending on the database size and the speed of your Internet connection.

2. Check the security package download status.

```
[edit]
user@host# run request security idp security-package download status
```

```
Done;Successfully downloaded from(https://signatures.juniper.net/cgi-bin/index.cgi).
Version info:2230(Mon Feb  4 19:40:13 2013 GMT-8, Detector=12.6.160121210)
```

3. Install the attack database.

```
[edit]
user@host# run request security idp security-package install
```

```
Will be processed in async mode. Check the status using the status checking CLI
```

NOTE: Installing the attack database might take some time depending on the security database size.

4. Check the attack database install status. The command output displays information about the downloaded and installed versions of the attack database.

```
[edit]
user@host# run request security idp security-package install status
```

```
Done;Attack DB update : successful - [UpdateNumber=2230,ExportDate=Mon Feb  4 19:40:13 2013
GMT-8,Detector=12.6.160121210]
Updating control-plane with new detector : successful
Updating data-plane with new attack or detector : successful
```

5. Confirm your IDP security package version.

```
[edit]
user@host# run show security idp security-package-version
```

```
Attack database version:2230(Mon Feb  4 19:40:13 2013 GMT-8)
  Detector version :12.6.160121210
  Policy template version :2230
```

6. Confirm your application identification package version.

```
[edit]
user@host# run show services application-identification version
```

```
Application package version: 1884
```

Verification

IN THIS SECTION

Confirm that the application signature package is being updated properly.

Verifying application signature package

Purpose

Verify the services application identification version.

Action

From operational mode, enter the `show services application-identification version` command.

```
user@host> show services application-identification version
```

```
Application package version: 1884
```

Meaning

The sample output shows that the services application identification version is 1884.

SEE ALSO

- request security idp security-package install
- request security idp security-package download
- Updating the IDP Signature Database Overview
- Understanding the IDP Signature Database

Downloading Junos OS Application Signature Package from A Proxy Server

IN THIS SECTION

This example shows how to create a proxy profile and use it for downloading the application signature package from a proxy server.

Configuration

Step-by-Step Procedure

Create a proxy profile and apply it for downloading the application package through the proxy server.

1. Create a proxy profile for protocol HTTP.

   ```
   user@host# set services proxy profile Profile-1 protocol http
   ```

2. Specify the IP address of the proxy server.

   ```
   user@host# set services proxy profile Profile-1 protocol http host 5.0.0.1
   ```

3. Specify the port number used by the proxy server.

   ```
   user@host# set services proxy profile Profile-1 protocol http port 3128
   ```

4. Download the application package from the proxy host.

   ```
   user@host# set services application-identification download proxy-profile Profile-1
   ```

Step-by-Step Procedure

You can disable the proxy server for downloading application signature package when not required.

- Disable the proxy server for application signature download.

  ```
  user@host# delete services application-identification download proxy-profile p1
  ```

Requirements

This example uses the following hardware and software components:

- Valid application identification feature license installed on an SRX Series Firewall.
- SRX Series Firewall with Junos OS Release 18.3R1 or later. This configuration example is tested for Junos OS Release 18.3R1.

Overview

You must download and install the application signature package that is hosted on an external server on the SRX Series Firewall. Starting from Junos OS Release 18.3R1, you can download the application signature package using a proxy server.

To enable downloading signature package from the proxy server:

1. Configure a profile with host and port details of the proxy server using the `set services proxy profile` command.

2. Use the `set services application-identification download proxy-profile` *`profile-name`* command to connect to the proxy server and download the application signature package.

When you download the signature package, the request is routed through the proxy host to the actual server hosting the signature package. The proxy host relays the response back from the actual host. The download retrieves the application package from the Juniper Networks security website https://signatures.juniper.net/cgi-bin/index.cgi.

NOTE: Support for the proxy profile configuration is available for only HTTP connections.

In this example, you create a proxy profile, and refer the profile when you download the application signature package from the external host. Table 2 on page 50 provides the details of the parameters used in this example.

Table 2: Proxy Profile Configuration Parameters

Parameter	Name
Profile Name	Profile-1
IP address of the proxy server	5.0.0.1
Port number of the proxy server	3128

Verification

IN THIS SECTION

Verifying Application Signature Download Through the Proxy Server

Purpose

Display the details for the application signature package download through a proxy server.

Action

From operational mode, enter the `show services application-identification status` command.

```
Application Identification
 Status                            Enabled
 Sessions under app detection      0
 Max TCP session packet memory     0
 Force packet plugin               Disabled
 Force stream plugin               Disabled
 DPI Performance mode:             Enabled
 Statistics collection interval    1440 (in minutes)

Application System Cache
 Status                            Enabled
  Cache lookup security-services   Enabled
  Cache lookup miscellaneous-services Enabled
 Max Number of entries in cache    131072
 Cache timeout                     3600 (in seconds)

Protocol Bundle
 Download Server                   https://signatures.juniper.net/cgi-bin/index.cgi
 AutoUpdate                        Disabled

Proxy Details
```

```
 Proxy Profile                     Profile-1
 Proxy Address                     http://5.0.0.1:3128
Slot 1:
 Application package version       3058
 Status                            Active
 PB Version                        1.340.0-57.005 (build date Apr 19 2018)
 Engine version                    4.20.0-91 (build date Feb 27 2018)
 Sessions                          0
```

Meaning

In the command output, you can find the proxy profile details in `Proxy Profile` and `Proxy Address` fields.

Verifying Application Signature Download Status

Purpose

Check the application package download status.

Action

From operational mode, enter the `request services application-identification download status` command.

```
user@host> request services application-identification download status
```

```
Application package 3058 is downloaded successfully
```

Meaning

The command displays the application signature package download status.

Example: Scheduling the Application Signature Package Updates

IN THIS SECTION

This example shows how to set up automatic updates of the predefined application signature package.

Requirements

Before you begin:

- Ensure that your security device has a connection to the Internet to download security package updates.

> **NOTE**: DNS must be set up because you need to resolve the name of the update server.

- Ensure that you have installed the application identification feature license.

Overview

In this example, you want to download the current version of the application signature package periodically. The download should start at 11:59 PM on December 10. To maintain the most current information, you want to update the package automatically every 2 days from your company's intranet site.

Configuration

IN THIS SECTION

Procedure

GUI Quick Configuration

To set up the automatic download and periodic update with the J-Web interface:

Step-by-Step Procedure

1. Enter `Configure>Security>AppSecure Settings` to display the Applications Signature page.

2. Click `Global Settings`.

3. Click the `Download Scheduler` tab, and modify the following fields:

   - URL: **https://signatures.juniper.net/cgi-bin/index.cgi**

   - Enable Schedule Update: Select the check box.

   - Interval: **48**

4. Click `Reset Setting` to clear the existing start time, enter the new start time in YYYY-MM-DD.hh:mm format, and click `OK`.

   - Start Time: **2019-06-30.10:00:00**

5. Click `Commit Options>Commit` to commit your changes.

6. Click `Check Status` to monitor the progress of an active download or update, or to check the outcome of the latest update.

Step-by-Step Procedure

To use the CLI to automatically update the Junos OS application signature package:

1. Specify the URL for the security package. The security package includes the detector and the latest attack objects and groups. The following statement specifies https://signatures.juniper.net/cgi-bin/index.cgi as the URL for downloading signature database updates:

   ```
   [edit]
   user@host# set services application-identification download url https://
   signatures.juniper.net/cgi-bin/index.cgi
   ```

2. Specify the time and interval for download. The following statement sets the interval as 48 hours and the start time as 10 am on December 10:

```
[edit]
user@host# set services application-identification download automatic interval 48 start-time
2019-06-30.10:00:00
```

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

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

Verification

To verify that the application signature package is being updated properly, enter the `show services application-identification version` command. Review the version number and details for the latest update.

SEE ALSO

- Understanding the Junos OS Application Package Installation | 34
- Downloading and Installing the Junos OS Application Signature Package Manually | 39
- Verifying the Junos OS Application Identification Extracted Application Package | 63

Scheduling the Application Signature Package Updates As Part of the IDP Security Package

IN THIS SECTION

The configuration instructions in this example describe how to setup automatic updates of application identification signature package (part of IDP security package) at a specified date and time.

Requirements

Before you begin:

- Ensure that your security device has a connection to the Internet to download security package updates.

  > **NOTE**: DNS must be set up because you need to resolve the name of the update server.

- Ensure that you have installed the application identification feature license.

Overview

In this example, you want to download the current version of the application signature package periodically. The download should start at 11:59 PM on December 10. To maintain the most current information, you want to update the package automatically every 2 days from your company's intranet site.

Configuration

IN THIS SECTION

Procedure

GUI Quick Configuration

To set up the automatic download and periodic update with the J-Web interface:

Step-by-Step Procedure

1. Enter `Configure>Security>IDP>Signature Updates` to display the Security IDP Signature Configuration page.

2. Click `Download Settings` and modify the URL: **https://signatures.juniper.net/cgi-bin/index.cgi**

3. Click the `Auto Download Settings` tab, and modify the following fields:

   - Interval: **48**

   - Start Time: **2013-12-10.23:59:55**

   - Enable Schedule Update: Select the check box.

4. Click `Reset Setting` to clear the existing fields, enter the new values. Click `OK`.

5. Click `Commit Options>Commit` to commit your changes.

6. Click `Check Status` to monitor the progress of an active download or update, or to check the outcome of the latest update.

Step-by-Step Procedure

To use the CLI to automatically update the Junos OS application signature package:

1. Specify the URL for the security package. The security package includes the detector and the latest attack objects and groups. The following statement specifies https://signatures.juniper.net/cgi-bin/index.cgi as the URL for downloading signature database updates:

   ```
   [edit]
   user@host# set security idp security-package url https://signatures.juniper.net/cgi-bin/
   index.cgi
   ```

2. Specify the time and interval for download. The following statement sets the interval as 48 hours and the start time as 11:55 pm on December 10, 2013:

   ```
   [edit]
   user@host# set security idp security-package automatic interval 48 start-time
   2013-12-10.23:55:55
   ```

3. Enable an automatic download and update of the security package.

   ```
   [edit]
   user@host# set security idp security-package automatic enable
   ```

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

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

Verification

IN THIS SECTION

Confirm that the application signature package is being updated properly.

Verifying application signature package

Purpose

Verify services application identification version

Action

From operational mode, enter the `show services application-identification version` command.

```
user@host> show services application-identification version
```

```
Application package version: 1884
```

Meaning

The sample output shows that, the services application identification version is 1884.

SEE ALSO

Understanding the Junos OS Application Package Installation | 34

Downloading and Installing the Junos OS Application Signature Package Manually | 39
Verifying the Junos OS Application Identification Extracted Application Package | 63

Example: Downloading and Installing the Application Identification Package in Chassis Cluster Mode

IN THIS SECTION

This example shows how to download and install the application signature package database to a device operating in chassis cluster mode.

Downloading and Installing the Application Identification Package

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 *CLI User Guide*.

To download and install an application package:

1. Download the application package on the primary node.

   ```
   {primary:node0}[edit]
   user@host> request services application-identification download
   ```

   ```
   Please use command "request services application-identification download status" to check
   status
   ```

2. Check the application package download status.

   ```
   {primary:node0}[edit]
   user@host> request services application-identification download status
   ```

On a successful download, the following message is displayed

```
Application package 2345 is downloaded successfully
```

The application package is installed in the application signature database on the primary node, and application identification files are synchronized on the primary and secondary nodes.

3. Update the application package using `install` command.

   `{primary:node0}[edit]`

   `user@host>` **request services application-identification install**

```
node0:
---------------------------------------------------------------------------
Please use command "request services application-identification install status" to check
status and use command "request services application-identification proto-bundle-status" to
check protocol bundle status

node1:
---------------------------------------------------------------------------
Please use command "request services application-identification install status" to check
status and use command "request services application-identification proto-bundle-status" to
check protocol bundle status
```

4. Check the application package update status. The command output displays information about the downloaded and installed versions of the application package.

   `{primary:node0}[edit]`

   `user@host>` **request services application-identification install status**

```
node0:
---------------------------------------------------------------------------
Install application package 2345 succeed

node1:
---------------------------------------------------------------------------
Install application package 2345 succeed
```

NOTE: It is possible that an application signature is removed from the new version of an application signature database. If this signature is used in an existing application firewall policy on your device, the installation of the new database will fail. An installation status message identifies the signature that is no longer valid. To update the database successfully, remove all references to the deleted signature from your existing policies and groups, and rerun the install command.

NOTE: While downloading the application signature package on the primary node, sometimes, due to unexpected failover, the primary node might not able to download the application signature package completely. As a workaround, you must delete the /var/db/appid/sec-download/.apppack_state and restart the device.

Step-by-Step Procedure

To uninstall an application package:

1. Uninstall the application package using `uninstall` command.

   `{primary:node0}[edit]`

   `user@host>` **request services application-identification uninstall**

   ```
   node0:
   --------------------------------------------------------------------------
   Please use command "request services application-identification uninstall status" to check
   status and use command "request services application-identification proto-bundle-status" to
   check protocol bundle status
   node1:
   --------------------------------------------------------------------------
   Please use command "request services application-identification uninstall status" to check
   status and use command "request services application-identification proto-bundle-status" to
   check protocol bundle status
   ```

2. Check the uninstall status of the application package.

   `{primary:node0}[edit]`

user@host> **request services application-identification uninstall status**

```
node0:
--------------------------------------------------------------------------
Uninstall application package 2345 succeed

node1:
--------------------------------------------------------------------------
Uninstall application package 2345 succeed
```

3. Check the uninstall status of protocol bundle:

```
user@host>request services application-identification proto-bundle-status
```

```
Protocol Bundle Version (1.30.4-22.005 (build date Jan 17 2014)) and application secpack
version (2345) is unloaded and deactivated
```

Requirements

Before you begin:

- Set the chassis cluster node ID and cluster ID. See *Example: Setting the Node ID and Cluster ID for Security Devices in a Chassis Cluster* .

- Ensure that your security device has a connection to the Internet to download security package updates.

> **NOTE**: DNS must be set up because you need to resolve the name of the update server.

- Ensure that you have installed application identification feature license.

Overview

If you use application identification, you can download the predefined application signature package database. Juniper Networks regularly updates the database and makes it available on the Juniper Networks website. This package includes application objects that can be used to match traffic in IDP, application firewall policies, and application tracking. For more details, see "Understanding the Junos OS Application Package Installation" on page 34.

When you download the application identification security package on a device operating in chassis cluster mode, the security package is downloaded to the primary node and then synchronized to the secondary node.

SEE ALSO

Understanding the Junos OS Application Package Installation | 34
Verifying the Junos OS Application Identification Extracted Application Package | 63

Verifying the Junos OS Application Identification Extracted Application Package

IN THIS SECTION

Purpose

After successful download and installation of the application package, use the following commands to view the predefined application signature package content.

Action

- View the current version of the application package:

```
show services application-identification version
```

```
Application package version: 1608
```

- View the current status of the application package:

```
show services application-identification status
```

```
pic: 1/0

Application Identification
 Status                            Enabled
 Sessions under app detection      0
 Engine Version                    4.18.1-20 (build date Jan 25 2014)
 Max TCP session packet memory     30000
 Max C2S bytes                     1024
 Max S2C bytes                     0
 Force packet plugin               Disabled
 Force stream plugin               Disabled
 Statistics collection interval    1 (in minutes)

Application System Cache
 Status                            Enabled
 Negative cache status             Disabled
 Max Number of entries in cache    131072
 Cache timeout in seconds          3600

Protocol Bundle
 Download Server                   https://signatures.juniper.net/cgi-bin/index.cgi
 AutoUpdate                        Enabled
Slot 1:
 Status                            Active
 Version                           1.30.4-22.005 (build date Jan 17 2014)
 Sessions                          0
Slot 2
 Status                            Free
```

SEE ALSO

Understanding the Junos OS Application Package Installation | 34

Downloading and Installing the Junos OS Application Signature Package Manually | 39

Uninstalling the Junos OS Application Identification Application Package

You can uninstall the predefined application package. The uninstall operation will fail if there are any active security policies referenced in the predefined application signatures in the Junos OS configuration

To uninstall application package:

1. Uninstall the application package:

   ```
   user@host> request services application-identification uninstall
   ```

   ```
   Please use command "request services application-identification uninstall status" to check
   status and use command "request services application-identification proto-bundle-status" to
   check protocol bundle status
   ```

2. Check the uninstall operation status of the application package. The command output displays information about the uninstall status of the application package and protocol bundle.

   - Check the uninstall status:

     ```
     user@host>request services application-identification uninstall status
     ```

     ```
     Uninstall application package 2345 succeed
     ```

   - Check the uninstall status of protocol bundle:

     ```
     user@host>request services application-identification proto-bundle-status
     ```

     ```
     Protocol Bundle Version (1.30.4-22.005 (build date Jan 17 2014)) and application secpack
     version (2345) is unloaded and deactivated
     ```

The application package and protocol bundle are uninstalled on the device. To reinstall application identification, you need to download application package and reinstall it again.

SEE ALSO

request services application-identification uninstall

request services application-identification uninstall status

Application Signature Package Rollback

IN THIS SECTION

Starting in Junos OS Release 20.3R1, you can rollback the current version of application signature pack to the previous version by one of the following methods:

- Automatic Rollback
- Manual Rollback

Automatic Rollback

In case of application signature package installation failure, the system automatically rolls back to the previous version of the application signature package that is currently installed on your security device.

When you download and install the application signature package on a device operating in chassis cluster mode, if the installation fails on a node, the system rolls back to the previous version of the application signature. The device displays a minor alarm on the same node where installation fails and rollback succeeds.

Example:

```
user@host> show system alarms

node0:
--------------------------------------------------------------------------
2 alarms currently active
Alarm time               Class  Description
2020-07-31 14:51:52 IST  Minor  APPIDD auto-rollback to previous version on install failure,
sigpack version on other node may differ
2020-07-31 13:23:26 IST  Minor  Rescue configuration is not set

node1:
--------------------------------------------------------------------------
1 alarms currently active
Alarm time               Class  Description
2020-07-31 13:23:23 IST  Minor  Rescue configuration is not set
```

Check application signature package rollback status when installation failed and the rollback completes successfully.

```
user@host>  request services application-identification rollback status
Application package rollback to version 3297 success
```

Manual Rollback

You can manually rollback the application signature package to the previous installed version using the following steps:

1. Rollback the application signature package to the previous version.

   ```
   user@host>  request services application-identification rollback
   Please use command        "request services application-identification rollback status" to
   check rollback status
   ```

2. Check the rollback status.

   ```
   user@host>  request services application-identification rollback status
   Application package rollback to version 3265 success.
   ```

Note the following for manual rollback of application signature package:

- Once you rollback application signature package version manually from version Y to version X, the scheduled auto-update of application signature package is skipped until a new version Z, which is higher than the version Y, is available.

- You can download and install application signatures through intrusion detection and prevention (IDP) security packages. In this case, if AppID installation fails during the IDP install, AppID rolls back to the previous version and IDP installation continues with the requested version. In such cases, IDP and AppID might have different versions.

- Application signature package installation does not proceed if there is any corruption, deletion, or modification of downloaded signature package files. In such cases, the following message is displayed:

```
user@host> request services application-identification install
error: Checksum validation failed for downloaded files.
```

- When your security device does not include any previous version application signature package and you attempt to rollback application signature package, the device displays the following error message:

```
user@host> request services application-identification install

No application package available to rollback.
```

Grouping Newly Added Application Signatures

IN THIS SECTION

Starting in Junos OS Release 21.1R1, we've enhanced application signature package by grouping all newly added application signatures under junos:all-new-apps group. When you download the application signature package on your security device, the entire predefined application group is downloaded and available for you to configure in security policy as shown in the below example:

```
user@host# set security policies from-zone untrust to-zone trust policy 1 match dynamic-
application
junos:all-new-apps
```

We've also introduced a list of application tags in the application signature package. You can group similar applications based on those predefined tags that are e based on application attributes. By doing so, you can consistently reuse the application groups when you define security policies.

```
user@host# set services application-identification application-group application-group-name tag-
group
tag-group-name applications-tags [web remote_access]
```

Example

```
user@host# set services application-identification application-group GROUP-1 tag-group TAG-1
application-tags [web remote_access]
```

```
user@host# set services application-identification application-group GROUP-1 tag-group TAG-2
application-tags [social_networking]
```

In the above example, you configure tag-based application group with tags remote-access and web and another tag group with social_networking. All the applications which are having tags as either web or remote-access and social_networking will be added to the application group.

Grouping of similar applications based on tags help you to consistently reuse the application groups when defining security policies.

Migration of New Applications to Normal Applications:

The junos:all-new-apps group contains a set of all new applications in the installed application signature pack on your security device compared to previously installed signature pack. If you decide to install a newer version of the application signature package, that version will contain a new set of applications in the junos:all-new-apps group.

You can chose to migrate the new applications to normal applications in your existing application signature package. This migration will help you to consistently maintain rules in security policy which are created specific to the new applications whenever you upgrade to newer application signature versions in future.

You can use the following new commands to move the applications tagged as new applications to normal applications:

- To migrate only specified new applications as normal application, use the following command:

```
request services application-identification new-to-production applications-list
[application-1 application-2]
```

- To migrate all new applications as normal applications, use the following command:

```
request services application-identification new-to-production all
```

After you run these commands, application will no longer be tagged as new and will not be part of the `junos:all-new-apps` group.

Application Signatures Package Enhancements

Starting in Junos OS Release 21.1R1, we've introduced the following enhancements to the application signature package:

- Support for FTP data context propagation
- Skipping of deep packet inspection (DPI) for the sessions offloaded by advanced policy-based routing (APBR) on application system cache (ASC) hit. (When only APBR service is enabled.)
- Forceful installation of the application signature pack over the same version of signature pack. See *request services application identification install ignore duplicate version check*
- Display of the application signature pack release date in the CLI command output. See *show services application-identification version*
- Display of the list of deprecated application signatures available in the installed signature pack in the CLI command output. See *show services application identification application obsolete applications*

NOTE: When you upgrade to Junos OS Release 21.1 and later from Junos OS Release 20.4 and earlier versions, we recommend you update the application identification signature database by using the `request services application-identifications download` and `request services application-identification install` commands.

Release History Table

Release	Description
20.4R1	Starting in Junos OS Release 20.4R1, system log messages are updated to display the application signature package download and installation results.
20.3R1	Starting in Junos OS Release 20.3R1, you can rollback the current version of application signature pack to the previous version
15.1X49-D65	Starting from 15.1X49-D65 and Junos OS Release 17.3R1, on SRX4100, and SRX4200 devices, AppSecure is part of Junos Software Enhanced (JSE) license package.
15.1X49-D40	Starting from Junos OS 15.1X49-D30 and Junos OS Release 17.3R1, on SRX300, SRX320, SRX340, and SRX345 devices, AppSecure is part of Junos Software Enhanced (JSE) software license package.
15.1X49-D30	Starting from Junos OS 15.1X49-D30 and Junos OS Release 17.3R1, on SRX1500 devices, AppSecure is part of Junos Software Enhanced (JSE) software license package.
12.1X47-D10	Starting from Junos OS Release 12.1X47-D10, next-generation application identification is supported.

RELATED DOCUMENTATION

Application Identification | 5

Custom Application Signatures for Application Identification | 72

Predefined and Custom Application Groups for Application Identification | 87

show services application identification application obsolete applications

Custom Application Signatures for Application Identification

IN THIS SECTION

User-defined custom application signatures can also be used to identify the application regardless of the protocol and port being used. You can create custom signatures using hostnames, IP address ranges, and ports, which allows you to track traffic to specific destinations. For more information, see the following topics:

Understanding Junos OS Application Identification Custom Application Signatures

IN THIS SECTION

This topic includes the following sections:

Custom Application Signatures Overview

Junos OS application identification feature provides you the flexibility to create custom signatures to identify any application, whether it is web-based or a client-server application. You can create custom application signatures for applications based on ICMP, IP protocol, IP address, and Layer 7.

In general, custom application signatures are unique to your environment and are mostly used to inspect internal or custom applications. Once you create custom application signatures, AppID classifies and inspects in the same manner as standard applications. Since custom application signatures are not part of the predefined application package, they are saved in the configuration hierarchy, not in the predefined application signature database.

You must download install the application signature package on your device to configure custom signatures. When the custom signatures are configured, you cannot uninstall the application signature package. All custom application signatures are carried forward as-is when you upgrade your system to a new software version.

Enhancements to Custom Application Signatures

Starting in Junos OS Release 20.1R1, we've enhanced the custom applications signature functionality by providing a new set of applications and contexts.

Custom application signature contexts are now part of application signature package. If you want to use the newly introduced application and contexts for custom application signatures, you must download and install the latest application signature package version 3248 or later. You can upgrade the application signature package separately without upgrading Junos OS.

Supported Types of Custom Application Signatures

Security devices support the following types of custom signatures:

- ICMP-based mapping
- Address-based mapping
- IP protocol-based mapping
- Layer 7-based and TCP/UDP stream-based mapping

In all supported custom application signatures, ICMP-based, IP protocol-based, and address-based custom applications have more priority than Layer 7-based and TCP/UDP stream based custom applications. Custom application signatures priority order is—ICMP-based, IP protocol-based, address-based, and Layer7-based or TCP/UDP stream-based custom applications.

ICMP-Based Mapping

- The ICMP mapping technique maps standard ICMP message types and optional codes to a unique application name. This mapping technique lets you differentiate between various types of ICMP messages. The ICMP mapping technique does not support ICMPv6 traffic.
- IDP works only with TCP or UDP traffic. Therefore, ICMP mapping does not apply to IDP and cannot support IDP features such as custom attacks.

Address-Based Mapping

- Layer 3 and Layer 4 address mapping defines an application by the IP address and optional port range of the traffic.
- For configuring Layer 3 and Layer 4 address-based custom applications, you must match the IP address and port range to destination IP address and port. When both IP address and port are configured, both criteria must match destination IP address and port range of the packet.

  Consider a Session Initiation Protocol (SIP) server that initiates sessions from its known port 5060. Because all traffic from this IP address and port is generated only by the SIP application, the SIP application can be mapped to the server's IP address and port 5060 for application identification. In this way, all traffic with this IP address and port is identified as SIP application traffic.
- When you configure an address-based application and a TCP/UDP stream-based application, and if a session matches both applications, the TCP/UDP stream-based application is reported as application and address-based application is reported as extended application.

⚠ **CAUTION**: To ensure adequate security, use address mapping when the configuration of your private network predicts application traffic to or from trusted servers. Address mapping provides efficiency and accuracy in handling traffic from a known application.

IP Protocol-Based Mapping

- Standard IP protocol numbers map an application to IP traffic. As with address mapping, to ensure adequate security, use IP protocol mapping only in your private network for the trusted servers.
- IDP works only with TCP or UDP traffic. IP protocol mapping, therefore, does not apply to IDP and cannot support IDP features such as custom attacks.

IP protocol based custom application signatures do not work as expected in Junos OS Releases in 19.2 through Junos OS Releases 19.4. Starting in Junos OS Release 20.1R1, you can use IP protocol-based custom application signatures.

Suggested workaround:

- If you are configuring unified policy, use service-based application configuration. Example:

  ```
  user@host# set applications application application-name protocol IP-proto-number
  ```

  Example:

  ```
  user@host# set applications application A1 protocol 2
  ```

- If you are using legacy application firewall, use predefined IP protocol applications. Example

  ```
  user@host# set security application-firewall rule-sets rule-set-name rule rule-name match
  dynamic-application application-name
  ```

  Example:

  ```
  user@host# set security application-firewall rule-sets RS-1 rule R1 match dynamic-application
  junos:IPP-IGMP
  ```

Layer 7-Based and TCP/UDP Stream-Based Signatures

- Layer 7 custom signatures define an application running over TCP or UDP or Layer 7 applications.
- Layer 7-based custom application signatures are required for the identification of multiple applications running on the same Layer 7 protocols. For example, applications such as Facebook and Yahoo Messenger can both run over HTTP, but there is a need to identify them as two different applications running on the same Layer 7 protocol.
- Layer 7-based custom application signatures detect applications based on the patterns in HTTP contexts. However, some HTTP sessions are encrypted in SSL. Application identification can also extract the server name information or the server certification from the TLS or SSL sessions. It can also detect patterns in TCP or UDP payload in Layer 7 applications.

Benefits of Using Custom Application Signatures

- Enforce security policy unique to your networking environment based on specific applications
- Bring visibility for unknown or unclassified applications
- Identify applications over Layer 7 and transiting or temporary applications, and to achieve further granularity of known applications

- Perform quality-of-service (QoS) for any specific application

Limitations

The following features are not supported:

- Some of the PCRE-based expressions and unicode-based characters (if not supported in Hyperscan)
- Enforcing of order among members in Layer 7-based signatures
- The wildcard address for address-based signatures (Layer 3 and Layer 4)

Additional Configuration Options for Custom Application Signatures

Starting in Junos OS Release 20.1R1 and if you are using application signature package version 3248 or later, you can configure the following options for custom application signatures:

Custom Application Pattern Depth

You can specify the byte limit for AppID to identify the custom application pattern for the applications running over TCP or UDP or Layer 7 applications.

To configure the limit, use the following configuration statements from the `[edit]` hierarchy:

```
user@host# set services application-identification application application-name over application
signature signature-name member number depth
```

Example:

```
user@host# set services application-identification application my_custom_address  over HTTP
signature my_addr_sig1 member m01 depth 256
```

For Layer 7 custom applications, the depth is considered from the beginning of the Layer 7 context. For TCP/UDP stream-based custom applications, depth is considered from the beginning of the TCP/UDP payload.

Custom Applications Inspection Byte Limit

You can set the inspection byte limit for AppID to conclude the classification and identify the custom application in a session. On exceeding the limit, AppID terminates the application classification. You can use this option to improve the application traffic throughput.

To configure the application byte limit, use the following configuration statements from the [edit] hierarchy:

```
user@host#  set services application-identification custom-application-byte-limit byte-number
```

Example:

```
user@host# set services application-identification custom-application-byte-limit 400
```

If you have configured a custom application signature over a predefined application and if AppID has already identified the predefined application, DPI continues with the custom signature identification. While the custom signature identification is in-progress, the classification is marked as non-final. If no custom application is identified within the custom application byte limit, and if predefined application is already identified, then AppID concludes the predefined application as final and offloads the session.

Priority for Custom Applications

In releases prior to Junos OS 20.1R1, the default priority for the custom application signatures was high which allowed custom signatures to take precedence over the predefined applications. Starting Junos OS release 20.1R1, the default priority for the custom application signature is low.

When AppID identifies a custom application with low priority before identifying a predefined application, it waits until predefined application classification is final. If there is no predefined application match available and the custom application is identified, then AppID terminates the classification with the identified custom application.

If you want to override the predefined applications priority with custom application signatures, you must explicitly set the priority to high for the custom application signatures.

To configure the high priority for custom applications, use the following configuration statements from the [edit] hierarchy:

```
user@host#  set services application-identification application application-name priority high
```

Example:

```
user@host#  set services application-identification application my_custom_address priority high
```

Note the following about priority of the custom applications:

- For Junos OS Release prior to 20.1R1:

  - The default priority for the custom applications is high.
  - The priority of the applications is considered when multiple applications match in the same packet.
  - When you configure high priority for custom application—Custom applications always have high precedence over the predefined applications.

    When you configure low priority for custom application—Custom applications have low precedence over similar pattern-based predefined signatures and high precedence over the other applications. In these releases, no option available to change the behavior.

- For Junos OS Release 20.1R1 and later:
  - The default priority for the custom applications is low.
  - The priority does not depend on the matches in the same packet.
  - The priority of Layer 7 and TCP/UDP stream based custom applications work as configured (either high or low) with all predefined applications.
  - Layer 3 and Layer 4 based custom applications always remains at high priority. In this case, the configured priority is ignored. Layer 3 and Layer 4 based custom applications override all predefined applications; because these applications are triggered on first packet of the session.

Subject Alternative Name

Starting in Junos OS Release 23.4R1, you can create an AppID custom signature using the SAN (Subject Alternative Name) certificate attribute for SSL signatures. An SSL certificate with the SAN attribute allows specifying multiple host names or IP addresses in a single certificate. With this enhancement, custom application signatures can detect applications based on the application's host names listed in the SAN field of the SSL certificate.

You can configure SAN using the `ssl-subject-alt-name` option under [`edit services application-identification application` *name* `over SSL signature` *name* `member` *name* `context`] hierarchy.

Example: Configuring Junos OS Application Identification Custom Application Signatures

IN THIS SECTION

This example shows how to configure custom application signatures for Junos OS application identification.

⚠ **CAUTION**: We recommend that only advanced Junos OS users attempt to customize application signatures.

Before You Begin:

- Install a valid application identification feature license on your SRX Series Firewall. See Managing Junos OS Licenses
- This configuration example is tested using Junos OS Release 20.1R1.
- Ensure that your security device with application signature package installed. See "Downloading and Installing the Junos OS Application Signature Package Manually" on page 39.
- To use enhanced custom application signatures, upgrade latest application signature package version 3284 or later. Check your application signature version using the following command:

```
user@host> show services application-identification version
```

```
Application package version: 3248
```

CAUTION: We recommend that only advanced Junos OS users attempt to customize application signatures.

Overview

Application identification supports custom application signatures to detect applications as they pass through the device. When you configure custom signatures, ensure that your signatures are unique.

Use the following steps to configure custom application signatures:

1. Define attributes such as context, patterns, direction, port range and so on for your security device to match the application traffic.

2. Configure inspection limit, pattern depth, and priority (optional configurations) to enhance custom applications application identification process.

3. Attach the custom application to a security policy that allows or denies the application traffic.

4. View application signatures and application signature groups by using the `show services application-identification application` and `show services application-identification group` commands.

Examples of Custom Application Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

- Set inspection limit for custom applications.

```
[edit ]
user@host# set services application-identification custom-application-byte-limit 400
```

- Set priority for custom applications.

```
[edit ]
user@host# set services application-identification application test cacheable
user@host# set services application-identification application test priority high
```

- Configure TCP stream-based custom signatures:

```
[edit ]
user@host# set services application-identification application my_custom_tcp over TCP
signature s1 member m01 context stream
user@host# set services application-identification application my_custom_tcp over TCP
signature s1 member m01 pattern .*install.*
user@host# set services application-identification application my_custom_tcp over TCP
signature s1 member m01 direction any
user@host# set services application-identification application my_custom_tcp over TCP
signature s1 member m01 depth 100
```

- Configure FTP context-based custom signatures:

```
[edit ]
user@host# set services application-identification application my_custom_ftp over FTP
signature sig1 member m01 depth 60
user@host# set services application-identification application my_custom_ftp over FTP
signature sig1 member m01 context ftp-file-name
user@host# set services application-identification application my_custom_ftp over FTP
signature sig1 member m01 pattern .*install.*
user@host# set services application-identification application my_custom_ftp over FTP
signature sig1 member m01 direction client-to-server
```

- Configure HTTP context-based custom signatures.

```
[edit ]
user@host# set services application-identification application my_custom_http over HTTP
signature s1 member m01 context http-header-host
user@host# set services application-identification application my_custom_http over HTTP
signature s1 member m01 pattern  .*agent1.*
user@host# set services application-identification application my_custom_http over HTTP
signature s1 member m01 direction client-to-server
```

```
user@host# set services application-identification application my_custom_http over HTTP
signature s1 member m01 depth 100
```

- Configure SSL context-based custom signatures:

```
[edit]
user@host# set  services application-identification application my_custom_ssl over SSL
signature s1 member m01 context ssl-server-name
user@host# set  services application-identification application my_custom_ssl over SSL
signature s1 member m01 pattern "example\.com"
user@host# set  services application-identification application my_custom_ssl over SSL
signature s1 member m01 direction client-to-server
user@host# set  services application-identification application my_custom_ssl over SSL
signature s1 member m01 depth 100
```

- Configure ICMP-based custom applications signatures:

```
[edit ]
user@host# set services application-identification application my_custom_icmp icmp-mapping
type 100
user@host# set services application-identification application my_custom_icmp icmp-mapping
code 1
```

- Configure Layer 3 or Layer 4 address-based custom applications signatures:

```
[edit ]
user@host# set services application-identification application my_custom_address address-
mapping ADDR-SAMPLE filter ip 192.0.2.1/24
user@host# set services application-identification application my_custom_address address-
mapping ADDR-SAMPLE filter port-range udp 5000-6000
```

NOTE: You must provide the appropriate port range and specified IP address to configure address-based custom application signatures.

- Configure IP protocol mapping-based custom application signatures.

```
[edit]
user@host# set  services application-identification application my_custom_ip_proto ip-
protocol-mapping protocol 2
```

- Create a security policy with custom applications as match criteria.

```
user@host# set security policies from-zone untrust to-zone trust policy 1 match source-
address any
user@host# set security policies from-zone untrust to-zone trust policy 1 match destination-
address any
user@host# set security policies from-zone untrust to-zone trust policy 1 match application
any
user@host# set security policies from-zone untrust to-zone trust policy 1 match dynamic-
application my_custom_http
user@host# set security policies from-zone untrust to-zone trust policy 1 then permit
```

  We are using my_custom_http for this example. Similarly, you can create different security policies and specify other custom applications such as my_custom_ftp, my_custom_tcp, my_custom_ssl, my_custom_address, my_custom_icmp, my_custom_ip_proto as match condition for the dynamic application as per your requirement.

- Enable application tracking.

```
user@host# set security zones security-zone trust application-tracking
```

Results

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

```
[edit]
user@host# show services application-identification
custom-application-byte-limit 100;
application my_custom_address {
    address-mapping ADDR-SAMPLE {
        filter {
            ip 192.0.2.1/24;
            port-range {
                udp 5000-6000;
            }
        }
    }
}
application my_custom_ftp {
    over FTP {
        signature sig1 {
            member m01 {
                depth 60;
                context ftp-file-name;
                pattern .*install.*;
                direction client-to-server;
            }
        }
    }
}
application my_custom_http {
    over HTTP {
        signature s1 {
            member m01 {
                depth 100;
                context http-header-host;
                pattern .*agent1.*;
                direction client-to-server;
            }
        }
    }
}
application my_custom_icmp {
    icmp-mapping {
        type 100;
        code 1;
    }
}
application my_custom_ip_proto {
    ip-protocol-mapping {
        protocol 2;
    }
}
application my_custom_ssl {
    over SSL {
        signature s1 {
            member m01 {
                depth 100;
                context ssl-server-name;
                pattern "example\.com";
                direction client-to-server;
            }
        }
    }
}
application my_custom_tcp {
    over TCP {
        signature s1 {
            member m01 {
                depth 100;
                context stream;
                pattern .*install.*;
                direction any;
            }
        }
    }
}
application test {
    cacheable;
    priority high;
}
```

```
[edit security policies]
user@host# show
from-zone untrust to-zone trust {
    policy 1 {
        match {
            source-address any;
            destination-address any;
            application any;
            dynamic-application [my_custom_http];
        }
        then {
            permit;
        }
    }
}
```

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

Verification

IN THIS SECTION

Verifying the Custom Application Definitions

Purpose

Display the custom application signatures configured on your device. Note that predefined application signature names use the prefix “junos:”

Action

From configuration mode, enter the `show services application-identification application detail` *`name`* command.

```
user@host> show services application-identification application
detail test

Application Name: test
Application type: TEST
Description: N/A
Application ID: 16777219
Priority: high
```

Meaning

The output of the command displays custom application name, type, description, ID, and the priority.

See *show services application-identification application*

SEE ALSO

Understanding the Junos OS Application Package Installation | 34

Customizing Application Groups for Junos OS Application Identification | 87

RELATED DOCUMENTATION

Application Identification | 5

Predefined Application Signatures for Application Identification | 33

Predefined and Custom Application Groups for Application Identification | 87

Predefined and Custom Application Groups for Application Identification

IN THIS SECTION

You can define an application group for both predefined applications, as well as custom applications. An application group contains applications that need similar treatment when defining a security policy. For more information, see the following topics:

Customizing Application Groups for Junos OS Application Identification

In Junos OS, application identification allows you to group applications in policies. Applications can be grouped under predefined and custom application groups. The entire predefined application group can be downloaded as part of the IDP or application identification security package. You can create custom application groups with a set of similar applications for consistent reuse when defining policies.

Application group support associates related applications under a single name for simplified, consistent reuse when using any application services.

As the predefined signature database changes, the content of a predefined application group can be modified to include new signatures

NOTE: An application group can contain applications and groups simultaneously. It is possible to assign one application to multiple groups. There is no limit to the number of dynamic application groups contained in one rule.

The hierarchy of application groups resembles a tree structure with associated applications as the leaf nodes. The group *any* refers to the root node. The group *unassigned* is always situated one level from the root and initially contains all applications. When a group is defined, applications are assigned from the unassigned group to the new group. When a group is deleted, its applications are moved back to the unassigned group.

All predefined application groups have the prefix “junos“ in the application group name to prevent naming conflicts with custom application groups. You cannot modify the list of applications within a predefined application group. However, you can copy a predefined application group to use it as a template for creating a custom application group.

To customize a predefined application group, you must first disable the predefined group. Note that a disabled predefined application group remains disabled after an application database update. You can then use the operational command `request services application-identification group` to copy the disabled predefined application group. The copied group is placed in the configuration file, and the prefix “junos” is changed to “my”. At this point, you can modify the list of applications in “my” application group and rename the group with a unique name.

To reassign an application from one custom group to another, you must remove the application from its current custom application group, and then reassign it to the other.

NOTE: Starting in Junos OS Release 18.2R2 and Junos OS Release 18.4R1, encrypted applications such as HTTP, SMTP, IMAP and POP3 over SSL are identified as junos:HTTPS, junos:SMTPS, junos:IMAPS, and junos:POP3S in Junos OS predefined applications and application sets.

For example: If you configure a security policy to allow or deny HTTPS traffic, you must specify application matching criteria as junos:HTTPS.

In previous Junos OS Releases, both HTTP and encrypted HTTP (HTTPS) applications can be configured using a same application matching criteria as junos:HTTP.

SEE ALSO

Understanding the Junos OS Application Identification Database | 9

Example: Configuring a Custom Application Group for Junos OS Application Identification for Simplified Management

IN THIS SECTION

This example shows how to configure custom application groups for Junos OS application identification for consistent reuse when defining policies.

Requirements

Before you begin, install an entire signature database from an IDP or an application identification security package. See "Downloading and Installing the Junos OS Application Signature Package Manually" on page 39 or "Downloading and Installing the Junos OS Application Signature Package As Part of the IDP Security Package" on page 44.

Overview

In this example, you define applications for an application group, delete an application from an application group, and include an application group within another application group.

In Junos OS, application identification allows you to group applications in policies. Applications can be grouped under predefined and custom application groups. The entire predefined application group can be downloaded as part of the IDP or application identification security package. You can create custom application groups with a set of similar applications for consistent reuse when defining policies.

NOTE: You cannot modify the applications defined in a predefined application group. However, you can copy a predefined application group using the operational command `request services` application-identification group *group-name* copy to create a custom application group and modify the list of applications. For more information, see request services application-identification group.

Configuration

IN THIS SECTION

Configuring Junos OS Application Identification User-Defined Application Groups

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 services application-identification application-group my_web
set services application-identification application-group my_web applications junos:HTTP
set services application-identification application-group my_web applications junos:FTP
set services application-identification application-group my_web applications junos:AMAZON
set services application-identification application-group my_web applications junos:GOPHER
set services application-identification application-group my_peer
set services application-identification application-group my_peer applications junos:BITTORRENT
set services application-identification application-group my_peer applications junos:BITTORRENT-
APPLICATION
set services application-identification application-group my_peer applications junos:BITTORRENT-
WEB-CLIENT
```

Step-by-Step Procedure

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

To configure a custom application group for application identification:

1. Set the name of your custom application group.

```
[edit services application-identification]
user@host# set application-group my_web
```

2. Add the list of applications that you want to include in your custom application group.

```
[edit services application-identification]
user@host# set application-group my_web applications junos:HTTP
user@host# set application-group my_web applications junos:FTP
user@host# set application-group my_web applications junos:GOPHER
user@host# set application-group my_web applications junos:AMAZON
```

3. Set the name of a second custom application group.

```
[edit services application-identification]
user@host# set application-group my_peer
```

4. Add the list of applications that you want to include in the group.

```
[edit services application-identification]
user@host# set application-group my_peer applications junos:BITTORRENT
user@host# set application-group my_peer applications junos:BITTORRENT-APPLICATION
user@host# set application-group my_peer applications junos:BITTORRENT-WEB-CLIENT
```

Results

From configuration mode, confirm your configuration by entering the `show services application-identification group` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show services application-identification application-group  my_web
    applications {
        junos:HTTP;
        junos:FTP;
        junos:GOPHER;
        junos:AMAZON
    }
user@host# show services application-identification application-group  my_peer
    applications {
        junos:BITTORRENT;
        junos:BITTORRENT-APPLICATION;
        junos:BITTORRENT-WEB-CLIENT;
    }
```

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

Deleting an Application from a User-Defined Application Group

CLI Quick Configuration

To quickly configure this section of the example, copy the following command, 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.

```
[edit]
delete services application-identification application-group my_web applications junos:AMAZON
```

Step-by-Step Procedure

To delete an application from a custom application group:

- Delete an application from a custom application group.

  ```
  [edit services application-identification]
  user@host# delete application-group my_web applications junos:AMAZON
  ```

1.

Results

From configuration mode, confirm your configuration by entering the `show services application-identification application group detail` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show services application-identification group detail
    application group my_web {
        junos:HTTP;
        junos:FTP;
        junos:GOPHER;
    }
```

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

Creating Child Application Groups for an Application Group

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 services application-identification application-group p2p
set services application-identification application-group p2p application-groups my_web
set services application-identification application-group p2p application-groups my_peer
```

Step-by-Step Procedure

To configure child application groups for a custom application group:

1. Set the name of the custom application group in which you are configuring the child application groups.

   ```
   [edit services application-identification]
   user@host# set application-group p2p
   ```

2. Add the child application groups.

```
[edit services application-identification]
user@host# set application-group p2p application-groups my_web
uer@host# set application-group p2p application-groups my_peer
```

Results

From configuration mode, confirm your configuration by entering the `show services application-identification application-group` *`application-group-name`* command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show services application-identification application-group p2p
    applications-groups {
        my_web;
        my_peer;
    }
```

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

SEE ALSO

Understanding Junos OS Application Identification Custom Application Signatures | 72

Enabling or Disabling Application Groups in Junos OS Application Identification

All application groups are enabled by default. Predefined application groups are enabled at installation.

- For predefined application groups, you can disable and reenable a group using the `request services application-identification group` command. You cannot delete a predefined signature or signature group.

- To disable a predefined application group:

```
user@host> request services application-identification group disable predefined-
application-group-name
```

NOTE: Make sure to commit the configuration changes or roll back the configuration when you are attempting to enable a disabled application or an application group. Uncommitted changes might result in configuration failure.

- To reenable a disabled predefined application group:

```
user@host> request services application-identification group enable predefined-application-
group-name
```

SEE ALSO

Understanding the Application System Cache | 11

RELATED DOCUMENTATION

Application Identification | 5
Predefined Application Signatures for Application Identification | 33
Understanding Junos OS Application Identification Custom Application Signatures | 72

Application Identification Support for Unified Policies

IN THIS SECTION

Understanding Unified Policies on Security Devices

IN THIS SECTION

With the growing popularity of Web applications, and because of the shift from traditional, full client-based applications to the Web, more and more traffic is being transmitted over HTTP. Applications such as instant messaging, peer-to-peer file sharing, Webmail, social networking, and IP voice and video collaboration evade security mechanisms by changing communication ports and protocols. Managing changes in the application behavior requires constant modification to the security rules, and maintenance of the security policy rules poses a major challenge. To handle such changes in application behavior, you need security policies to manage dynamic applications.

As a response to this challenge, starting in Junos OS Release 18.2R1, Juniper Networks SRX Series Firewalls and vSRX Virtual Firewall support *unified policies*, allowing granular control and enforcement of dynamic Layer 7 applications within the security policy. Unified policies are security policies that enable you to use dynamic applications as part of the existing 5-tuple or 6-tuple (5-tuple with user firewall) match conditions to detect application changes over time.

A unified policy leverages the application identity information determined from the application identification (AppID) module. After a particular application is identified, an action such as permit, deny, reject, or redirect is applied to the traffic according to the policy configured on the device.

Any traffic denied or rejected by the security policy based on Layer 3 or Layer 4 criteria is dropped immediately. Traffic permitted by the security policy is further assessed at Layer 7 based on its AppID information.

AppID is enabled when you configure a security policy with dynamic applications or when you enable any services such as application policy-based routing (APBR), application tracking (Apptrack), application quality of service (AppQoS), application firewall (AppFW), IDP, or Juniper ATP Cloud in the security policy.

Benefits

- Simplifies application-based security policy management at Layer 7.
- Enables your device to adapt to the dynamic traffic changes in the network.
- Provides greater control and extensibility to manage dynamic applications traffic than a traditional security policy.

Understanding How Unified Policies Use AppID Information

IN THIS SECTION

Accurate traffic classification is essential for network security in cloud and data center architectures. Identifying and classifying different types of application traffic (transacted on HTTP) is also a challenge as Web applications include documents, data, images, and audio and video files.

AppID detects the applications on your network regardless of the port, protocol, and encryption (TLS/SSL or SSH) or other evasive tactics. It uses deep packet inspection (DPI) techniques, a signature database, and well-known addresses and ports to identify applications. AppID provides the information such as dynamic application classification, default protocol and port of an application. For any application that is included in the dependent list of another application, AppID provides the information of dependent application.

A unified policy leverages the information from AppID to match the application and take action as specified in the policy. In a unified policy configuration, you can use a predefined dynamic application (from the application identification signature package) or a user-defined custom application as match condition.

Understanding Dependent Dynamic Application Identification

A dependent application list includes applications over which a dynamic application can be identified. For example, the dependent application list for Facebook comprises HTTP2 and SSL.

The default protocol and port of a dynamic application includes the protocol and port defined for that application. If the protocol and port for that application is not defined, then the list of default protocols and ports of its dependent applications is considered.

For example, the Facebook-Access application depends on applications such as HTTP, SSL, and HTTP2. Therefore, the default protocol and ports of these dependent applications are considered for the Facebook-Access application.

NOTE: The dependent application list and protocol and port mapping of an application might change during runtime whenever a new application signature pack is installed or a custom application configuration changes. AppID provides these details to the security policy.

Dynamic Application Classification States

During the application identification process, DPI processes every packet and classifies it into one of the following states until the application is finally identified:

- Pre-match—Before an application is identified by the DPI.

- Transaction final—For dynamic applications, one transaction is complete, but identification of the application is not final. Applications over Layer 7 can keep changing with each transaction because they have dependent applications. For example, Facebook applications have dependent applications such as HTTP, SSL, and so on.

- Final match—A matched application over Layer 7 is considered as the final match according to the configured maximum number of transactions. That is, the match is considered as final only after the maximum number of transactions are complete.

Before identifying the final application, the policy cannot be matched precisely. A potential policy list is made available, and the traffic is permitted using the potential policy from the list. After the application is identified, the final policy is applied to the session. Policy actions such as permit, deny, reject, or redirect are applied to the traffic as specified in the policy rules.

Application classification is not terminated for applications that are transaction-based, such as Facebook. To terminate the classification for such applications, you can choose to consider the results from multiple transactions as the final classification.

Configuring Transactions Limit For Application Identification

You can configure the maximum number of transactions before concluding the final results for identifying an application using the **set services application-identification maximum-transactions *transactions-number*** statement. When you configure the maximum number of transactions, DPI is not terminated until the configured number of transactions are completed.

Example:

```
user@host# set services application-identification maximum-transactions 5
```

You can configure a transaction number from 0 through 25. By default, five transactions are considered.

If you set the transaction count as 0, the transaction does not terminate the DPI. The final match for the application might not be available; and the final security policy is not applied.

Table 3 on page 99 shows the different states of application identification classification when the maximum transaction is set as five. Note that the values in the table are for example and are not actual values. The exact transaction might vary depending on the traffic pattern.

Table 3: Application Identification Transactions Example

Scenario	Application Identified	Application Identification State	Transactions
First packet of the session	None	Pre-match	0
Intermediate application	SSL	Pre-match	1
Intermediate application identified in decrypted payload	HTTP	Pre-match	2
Intermediate application identified	FACEBOOK-ACCESS	Pre-match	3
Intermediate application identified	FACEBOOK-CHAT	Final Transaction (Transaction =1)	4
Final application identified	FACEBOOK-MAIL	Final Match (Transaction = 2)	4

NOTE: In unified policies, configuring dynamic applications that can be identified based on Layer 3 or Layer 4 information (except ICMP-based applications) is not supported. Instead, you can use the junos-defaults group that contains predefined values for Layer 3 and Layer 4 based applications.

High Availability Support for Application Identification for Unified Policies

When an application is identified, its classification information is saved in the application system cache (ASC).

When your security device (example: SRX Series Firewall) is operating in chassis cluster mode, the information saved in the ASC is synchronized between the primary node and the secondary node.

In case of dynamic application classification, per session application classification information from the DPI is synchronized with the secondary node when the application classification is final.

During a failover, the application classification information on the secondary node is in either of the following states:

- Application not identified
- Final application identified

After a failover, the application classification information that is available in the new primary node is considered as the final match. The same information is synchronized with the new secondary node as the classification does not proceed further after a failover. The example in Table 2 Table 4 on page 100 shows application classification status in a chassis cluster setup.

Table 4: Application Classification Status in a Chassis Cluster Setup

Application Identification Status	Chassis Cluster Node	Before Failover	After Failover	Details
Final application is identified. Identified application: SSL:Facebook	Primary node	Identified application: SSL:Facebook	Identified application: SSL:Facebook	No change after failover because complete application classification is synchronized to the secondary node.
	Secondary node	Identified application: SSL:Facebook	Identified application: SSL:Facebook	

Table 4: Application Classification Status in a Chassis Cluster Setup *(Continued)*

Application Identification Status	Chassis Cluster Node	Before Failover	After Failover	Details
Final application is not identified. (Partial application is identified.) Identified application: SSL	Primary node	Identified application: SSL	Identified application: APP-INVALID	Application identification does not proceed further after a failover.
	Secondary node	Identified application: not available	Identified application: APP-INVALID	
Final application is not identified. (Partial application is identified)	Primary node	Identified application: not available	Identified application: APP-INVALID	In this case, a failover occurred after the first packet inspection, and no application is identified. Application identification does not proceed further after a failover.
	Secondary node	Identified application: not available	Identified application: APP-INVALID	

Enabling or Disabling Application System Cache for Application Services

Starting in Junos OS Release 18.2R1, the default behavior of the ASC is changed as follows:

- Before Junos OS Release 18.2R1—ASC is enabled by default for all services including security services.
- From Junos OS Release 18.2R1 onwards—ASC is enabled by default; note the difference in security services lookup:
  - ASC lookup for security services is not enabled by default. That is—security services including security policies, application firewall (AppFW), application tracking (AppTrack), application quality of service (AppQoS), Juniper ATP Cloud, IDP, and Content Security do not use the ASC by default.

- ASC lookup for miscellaneous services is enabled by default. That is—miscellaneous services including advanced policy-based routing (APBR) use the ASC for application identification by default.

NOTE: The change in the default behavior of the ASC affects the legacy AppFW functionality. With the ASC disabled by default for the security services starting in Junos OS Release 18.2 onward, AppFW will not use the entries present in the ASC.

You can revert to the ASC behavior as in Junos OS releases before Release 18.2 by using the `set services application-identification application-system-cache security-services` command.

CAUTION: The security device might become susceptible to application evasion techniques if the ASC is enabled for security services. We recommend that you enable the ASC only when the performance of the device in its default configuration (disabled for security services) is not sufficient for your specific use case.

Use the following commands to enable or disable the ASC:

- Enable the ASC for security services:

```
user@host# set services application-identification application-system-cache security-services
```

- Disable the ASC for miscellaneous services:

```
user@host# set services application-identification application-system-cache no-miscellaneous-services
```

- Disable the enabled ASC for security services:

```
user@host# delete services application-identification application-system-cache security-services
```

- Enable the disabled ASC for miscellaneous services:

```
user@host# delete services application-identification application-system-cache no-miscellaneous-services
```

You can use the `show services application-identification application-system-cache` command to verify the status of the ASC.

The following sample output provides the status of the ASC:

```
user@host>show services application-identification application-system-cache
Application System Cache Configurations:
  application-cache: on
    Cache lookup for security-services: off
    Cache lookup for miscellaneous-services: on
  cache-entry-timeout: 3600 seconds
```

In releases before Junos OS Release 18.2R1, application caching was enabled by default. You can manually disable it by using the `set services application-identification no-application-system-cache` command.

```
user@host# set services application-identification no-application-system-cache
```

SEE ALSO

- Understanding Application Identification Techniques | 5
- Verifying Application System Cache Statistics | 13
- Understanding the Junos OS Application Identification Database | 9

Tunnelling Applications Support

Starting in Junos OS Release 20.4R1, we've enhanced unified policy lookup on security device to manage tunneling applications. You can now block a specific tunneling application by using the unified policy.

When you want to block certain tunneling applications such as QUIC or SOCK, you can configure these tunneling applications to unified policy with action deny or reject.

Application Identification Support for Micro-Applications

IN THIS SECTION

Starting in Junos OS Release 19.2R1 onwards, you can manage the applications at a sub-function level with application identification feature. In this document, we refer application sub-functions as micro-applications.

Micro-applications are part of application signature package. You must enable micro-application detection in application identification and then use them as matching criteria in security policy.

AppID detects the applications at sub-function level on your network and security policy leverages the application identity information determined from the application identification (AppID) module. After a particular application is identified, an action such as permit, deny, reject, or redirect is applied to the traffic according to the policy configured on the device.

Micro-applications concept is similar to transaction-based applications, where the nested application over a base application continuously change for the same session.

Example:

Consider a dynamic application MODBUS. READ and WRITE are sub functions or operations of MODBUS application. For these sub-functions, we must define micro-applications such as MODBUS-READ and MODBUS-WRITE. Application classification path can keep changing between MODBUS:MODBUS-READ and MODBUS:MODBUS-WRITE. In this case, MODBUS is the base application and MODBUS-READ and MODBUS-WRITE are nested applications, that is, micro-applications.

You can configure the micro-applications at the same hierarchy as predefined dynamic application in a security policy and take the action based on the policy rules.

By configuring these micro-applications in security policies, you can allow or deny MODBUS sub-functions rather than blocking or allowing the entire MODBUS application.

Micro-Application Classification

Application classification for micro-applications does not reach to the final match because, the micro-application keep changing for the session. A matched application is considered as the final match only after the maximum number of transactions are complete.

AppID has the maximum transaction limit as 25, however each service module has it's own limit based on it's own requirements. If service specific limit is reached before the maximum transaction limit (25), then the service module marks it's policy as final. However, AppID continues application classification and offloads the session on reaching the limit of 25.

You can use the `set services application-identification max-transactions` command to configure the transaction limit.

Dependent Application List and Default Protocols and Ports

A dependent application list includes applications over which a dynamic application can be identified. The default protocol and port of a dynamic application includes the protocol and port defined for that application.

Dependent application list and default protocols and ports are used by unified policy for enforcing the security policy. Dependent application list and default protocols and ports of micro application is same as that of base application.

Example: Dependent application list and default ports of micro-application MODBUS-READ is same as dependent application list and default ports of MODBUS.

Policy Enforcement for Micro-Applications

A security policies enforce rules for transit traffic, in terms of what traffic can pass through the device, and the actions that need to take place on traffic as it passes through the device. If you have configured a security policy with micro-application as match criteria, then the policy module requires micro-application identification information from AppID.

Application classification with micro-applications does not reach the final match because, the micro-application keep changing for the session. However, final match for the application is required for policy lookup and processing of the policy. You can use the [`edit security policies unified-policy-max-lookups`] command to limit the number of policy lookups.

After the application is identified, the final policy is applied to the session. Policy actions such as permit, deny, reject, or redirect are applied to the traffic as specified in the policy rules.

Installing Micro-Applications

Micro applications are part of application signature package. When you download application signature package and install it, micro applications are also installed and are available for configuring in the security policies. You can view the details of the micro applications using the `show services application-identification status` command.

> **NOTE**: If you have configured micro-applications in a security policy starting in Junos OS Release 19.2, it is not possible to downgrade to the previous version of Junos OS release. To downgrade to the previous version of Junos OS releases, you must remove the micro applications configured in your security policies.

Managing DNS-over-HTTP and DNS-over-TLS Application Traffic

In Junos OS Release 20.4R1, we introduce a new micro-application, DNS-ENCRYPTED, to enhance the application signature package. By configuring this micro-application in a security policy, you can have granular control for DNS-over-HTTP and DNS-over-TLS application traffic.

The DNS-ENCRYPTED application is enabled by default. You can disable it using the `request services application-identification application disable DNS-ENCRYPTED` command.

You can view the details of the micro-applications using the `show services application-identification application` command.

Enabling and Disabling Micro-Applications Detection

You can enable or disable micro-application detection. By default, detection of micro-applications are disabled. You must enable micro-applications to use them in your security policy.

You can enable or disable micro-applications using the following commands:

- Enable micro-applications detection (from configuration mode).

  ```
  user@host# set services application-identification micro-apps
  ```

- Disable a specific micro-application (from operational mode).

  ```
  user@host> request services application-identification application disable application-name
  ```

Example:

```
user@host>request services application-identification application disable junos:MODBUS
```

Example: Configuring Micro-Applications

IN THIS SECTION

This example shows how to configure micro-applications in a security policy to enforce the policy at sub-function level.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall with Junos OS Release 19.2R1 or later. This configuration example is tested on Junos OS Release 19.2R1.
- Valid application identification feature license installed on an SRX Series Firewall.

Before you begin, install an entire signature database from an IDP or an application identification security package. See "Downloading and Installing the Junos OS Application Signature Package Manually" on page 39 or "Downloading and Installing the Junos OS Application Signature Package As Part of the IDP Security Package" on page 44.

Overview

In this example, you create a security policy with micro-applications MODBUS-READ-COILS and MODBUS-WRITE-SINGLE-COIL, MODBUS-READ-COILS, MODBUS-WRITE-MULTIPLE-COILS. Application traffic matching these micro-applications is permitted.

Configuration

IN THIS SECTION

Configuring Security Policy with Micro-Applications

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 services application-identification micro-apps
set security policies from-zone untrust to-zone trust policy P1 match source-address any
set security policies from-zone untrust to-zone trust policy P1 match destination-address any
set security policies from-zone untrust to-zone trust policy P1 match application any
set security policies from-zone untrust to-zone trust policy P1 match dynamic-application
junos:MODBUS-READ-COILS
set security policies from-zone untrust to-zone trust policy P1 match dynamic-application
junos:MODBUS-WRITE-SINGLE-COIL
set security policies from-zone untrust to-zone trust policy P1 match dynamic-application
junos:MODBUS-WRITE-MULTIPLE-COILS
set security policies from-zone untrust to-zone trust policy P1 then permit
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure a custom application group for application identification:

1. Enable micro-applications detection.

   ```
   [edit]
   user@host# set services application-identification micro-apps
   ```

2. Define a security policy with other policy matching criteria.

   ```
   [edit]
   user@host# set security policies from-zone untrust to-zone trust policy P1 match source-
   address any
   user@host# set security policies from-zone untrust to-zone trust policy P1 match destination-
   address any
   user@host# set security policies from-zone untrust to-zone trust policy P1 match application
   any
   ```

3. Define application and micro-application as matching criteria.

   ```
   [edit]
   user@host# set security policies from-zone untrust to-zone trust policy P1 match dynamic-
   application  junos:MODBUS-READ-COILS
   user@host# set security policies from-zone untrust to-zone trust policy P1 match dynamic-
   application  junos:MODBUS-WRITE-SINGLE-COIL
   user@host# set security policies from-zone untrust to-zone trust policy P1 match dynamic-
   application  junos:MODBUS-WRITE-MULTIPLE-COILS
   ```

4. Define the policy action.

   ```
   [edit]
   user@host# set security policies from-zone untrust to-zone trust policy P1 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.

```
[edit]
user@host# show security policies from-zone untrust to-zone trust
```

```
from-zone untrust to-zone trust {
    policy P1 {
        match {
            source-address any;
            destination-address any;
            application any;
            dynamic-application [ junos:MODBUS-READ-COILS junos:MODBUS-WRITE-SINGLE-COIL
junos:MODBUS-WRITE-MULTIPLE-COILS ];
        }
        then {
            permit;
        }
    }
}
```

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

Configuring Application Quality-of-Service with Micro-Applications

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy.

To configure a custom application group for application identification:

1. Define AppQoS configuration parameters with micro-application junos:MODBUS-READ-COILS.

   ```
   [edit]
   user@host# set class-of-service application-traffic-control rate-limiters RL1 bandwidth-limit
   1000
   user@host# set class-of-service application-traffic-control rule-sets RS1 rule 1 match
   application junos:MODBUS-READ-COILS
   user@host#  set class-of-service application-traffic-control rule-sets RS1 rule 1 then dscp-
   code-point 111110
   user@host# set class-of-service application-traffic-control rule-sets RS1 rule 1 then loss-
   priority high
   user@host# set class-of-service application-traffic-control rule-sets RS1 rule 1 then rate-
   limit client-to-server RL1
   user@host# set class-of-service application-traffic-control rule-sets RS1 rule 1 then log
   ```

2. Create a security policy.

```
[edit security]
user@host# set security policies from-zone untrust to-zone trust policy 1 match source-address any
user@host#  set security policies from-zone untrust to-zone trust policy 1 match destination-address any
user@host# set security policies from-zone untrust to-zone trust policy 1 match application any
```

3. Define the policy action.

```
[edit security]
user@host# set security policies from-zone untrust to-zone trust policy 1 then permit application-services application-traffic-control rule-set RS1
```

Results

From configuration mode, confirm your configuration by entering the `how class-of-service` command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service
application-traffic-control {
    rate-limiters RL1 {
        bandwidth-limit 1000;
    }
    rule-sets RS1 {
        rule 1 {
            match {
                application junos:MODBUS-READ-COILS;
            }
            then {
                dscp-code-point 111110;
                loss-priority high;
                rate-limit {
                    client-to-server RL1;
                }
                log;
            }
        }
    }
}
```

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

```
[edit]
user@host# show security policies from-zone untrust to-zone trust
from-zone untrust to-zone trust {
    policy 1 {
        match {
            source-address any;
            destination-address any;
            application any;
            dynamic-application [ junos:MODBUS-READ-COILS];
        }
        then {
            permit {
                application-services {
                    application-traffic-control {
                        rule-set RS1;
                    }
                }
            }
        }
    }
}
```

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

Verification

IN THIS SECTION

Verifying Micro-Applications Status

Purpose

Verify that micro-applications are enabled.

Action

Use the `show services application-identification status` command to get micro-applications version and use `show services application-identification application micro-applications` command to get the details of the micro-applications.

```
Application Identification
 Status                            Enabled
 Sessions under app detection      0
 Max TCP session packet memory     0
 Force packet plugin               Disabled
 Force stream plugin               Disabled
 Statistics collection interval    1440 (in minutes)

Application System Cache
 Status                            Enabled
  Cache lookup security-services   Disabled
  Cache lookup miscellaneous-services Disabled
 Max Number of entries in cache    0
 Cache timeout                     3600 (in seconds)

Protocol Bundle
 Download Server                   https://signatures.juniper.net/cgi-bin/index.cgi
 AutoUpdate                        Disabled

Proxy Details
 Proxy Profile                     Not Configured
Slot 1:
 Application package version       3172
 Status                            Active
 PB Version                        1.380.0-64.005 (build date May 13 2019)
 Engine version                    5.3.0-56 (build date May 13 2019)
 Micro-App Version                 1.0.0-0
 Sessions                          0
```

Sample Output

show services application-identification application micro-applications

```
user@host> show services application-identification application micro-applications



Micro Applications
  junos:BACNET-GET-EVENT-INFORMATION
  junos:BACNET-SUBSCRIBE-COV-PROPERTY
  junos:BACNET-LIFE-SAFETY-OPERATION
  junos:BACNET-READ-RANGE
  junos:BACNET-REQUEST-KEY
  junos:BACNET-AUTHENTICATE
  junos:BACNET-VT-DATA
  junos:BACNET-VT-CLOSE
  junos:BACNET-VT-OPEN
  junos:BACNET-REINITIALIZE-DEVICE
  junos:BACNET-CONFIRMED-TEXT-MESSAGE
  junos:BACNET-CONFIRMED-PRIVATE-XFER
  junos:BACNET-DEVICE-COMM-CONTROL
  junos:BACNET-WRITE-PROP-MULTIPLE
  junos:BACNET-WRITE-PROPERTY
  junos:BACNET-READ-PROP-MULTIPLE
  junos:BACNET-READ-PROP-CONDITIONAL
  junos:BACNET-READ-PROPERTY
  junos:BACNET-DELETE-OBJECT
  junos:BACNET-CREATE-OBJECT
  junos:BACNET-REMOVE-LIST-ELEMENT
  junos:BACNET-ADD-LIST-ELEMENT
  junos:BACNET-ATOMIC-WRITE-FILE
  junos:BACNET-ATOMIC-READ-FILE
  junos:BACNET-SUBSCRIBE-COV
  junos:SIEMENS-S7-SETUP-COMM
  junos:SIEMENS-S7-UPLOAD-START
......
```

See *show services application-identification application micro-applications* for more details.

Verifying Micro-Applications Statistics

Purpose

Verify that micro-application are applied.

Action

Use the following commands to get the details of the micro-applications.

Sample Output

command-name

```
user@host> show services application-identification statistics applications



Last Reset: 2018-12-16 01:45:47 PST
     Application              Sessions          Bytes    Encrypted
      MODBUS-READ-COILS               1           1026           No
      MODBUS-WRITE-SINGLE-COIL        1           1254           No
```

```
user@host> show services application-identification statistics applications details (Junos OS
Release 20.3)

Logical System: root-logical-system
Last Reset: 2020-05-08 08:55:31 PDT
            Application  Enc    DPI final-match    Pre-match  Limits
                                                               final-match
                    NTP   No                  1            0            0
                 SYSLOG   No                  5            0            0
```

SEE ALSO

- show services application-identification application micro-applications
- show services application-identification application non-configurable

Understanding Junos OS Application Identification Custom Application Signatures

Release History Table

Release	Description
18.2R1	Starting in Junos OS Release 18.2R1, the default behavior of the ASC is changed
18.2R1	In releases before Junos OS Release 18.2R1, application caching was enabled by default. You can manually disable it by using the `set services application-identification no-application-system-cache` command.

RELATED DOCUMENTATION

Application Identification | 5

Secure Web Proxy

SUMMARY

You can use a Juniper Networks SRX Series Firewall to configure *secure Web proxy* to selectively bypass the external proxy server for the traffic based on application types. Read this topic to understand how secure Web proxy works and how you can configure it on your SRX Series Firewall.

IN THIS SECTION

Secure Web Proxy Overview

IN THIS SECTION

You can use secure Web proxy to send traffic to an external proxy server and bypass the proxy server for the selected application traffic. Bypassed application traffic will be sent directly to the target webserver.

To use secure Web proxy on an SRX Series Firewall, you must configure a secure Web proxy profile with external proxy server details and dynamic application that you want to bypass the external proxy server. When the security device receives a request from a client, the device examines the HTTP header for the application. The device applies Web proxy profile for the traffic that matches the security policy rules. Permitted application traffic that matches the dynamic-application specified in the Web proxy profile, is directed to the webserver. Otherwise, the permitted traffic is re-directed to the configured external proxy server.

As a result, your security device performs transparent proxy between the client and the webserver for the specified applications and provides better quality of service for the application traffic.

Starting in Junos OS Release 19.2R1, you can configure secure Web proxy on the following SRX Series Firewalls—SRX300, SRX320, SRX340, SRX345, SRX550, SRX1500, SRX4100, SRX4200, and vSRX Virtual Firewall.

Benefit

- Secure Web proxy provides better quality of service for the selected application traffic by providing direct connections to the webserver

Limitations

- An SRX Series Firewall operating in chassis cluster mode does not support the secure Web proxy functionality.

- Advanced policy-based routing (APBR), when applied along with secure Web proxy, works fine. However, other Layer 7 services might not work along with Secure Web Proxy as expected.

- If you have configured unified policies (security policies with dynamic applications) on your SRX Series Firewall, the Secure Web Proxy feature may not function properly.

- Secure Web proxy feature is not supported when device is operating in transparent-bridge mode.

- Secure Web Proxy feature does not work when the client device and its proxy server are deployed in the same network segment.

How Secure Web Proxy Works on SRX Series Firewalls

Figure 3 on page 118 and Figure 5 on page 119 show how an SRX Series Firewall provides the secure Web proxy service.

Figure 3: Secure Web Proxy on SRX Series Firewall

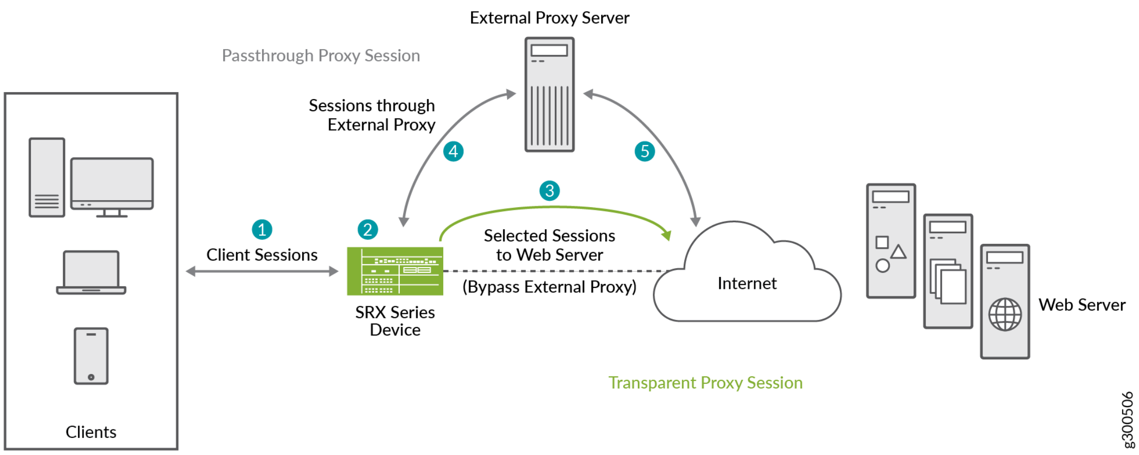


Figure 4: Secure Web Proxy on SRX Series Firewall—Workflow

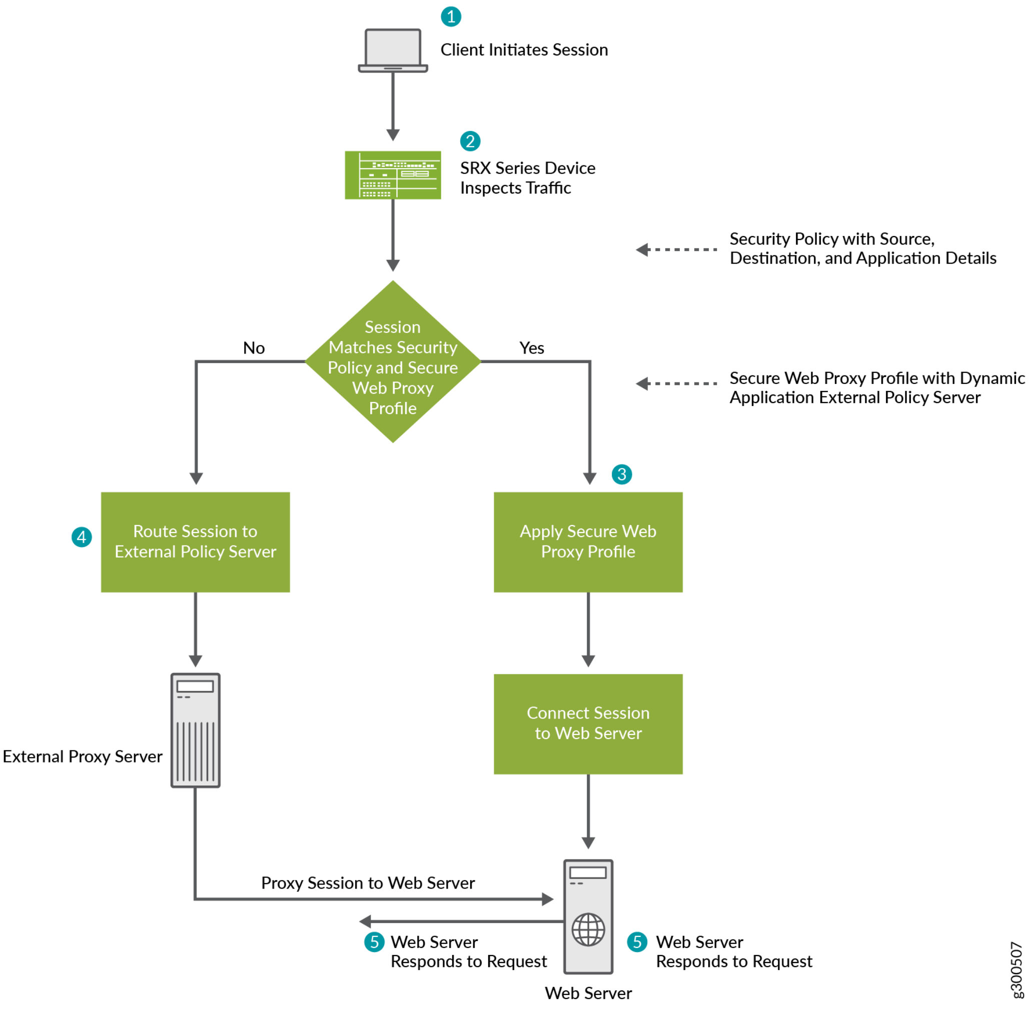


Figure 5: Workflow - Secure Web Proxy on SRX Series Firewall

To use secure Web proxy on your SRX Series Firewall, you must:

1. Create a secure Web proxy profile, which includes the details about the external proxy server and the dynamic application or application group that can bypass the external proxy server.

2. Create a security policy to manage the traffic passing through the device.

3. Attach the secure Web proxy profile to the security policy and apply the profile as an application service for the permitted traffic.

When a client initiates a request, the SRX Series Firewall examines the application traffic and identifies which traffic can bypass the external proxy server based on the secure Web proxy profile and security policy rules.

For example, if you use Microsoft Office 365, you can specify an Office 365 application group, such as junos:OUTLOOK or junos:OFFICE365-CREATE-CONVERSATION, in the secure Web proxy profile. The SRX Series Firewall forwards the Office 365 application traffic directly to the Office 365 server, bypassing the external proxy server. Connections that do not match the applications are routed to the external proxy server.

The SRX Series Firewall performs secure Web proxy through the following steps:

1. The client's browser sends an HTTP connect request to the external proxy server.
2. The SRX Series Firewall intercepts the TCP connections. The device identifies the application in the HTTP header and does a DNS resolution.
3. If the traffic parameters match the security policy rules and the secure Web proxy profile specifications, the SRX Series Firewall operates in transparent mode. The device uses the client's IP address in transparent mode to initiate a new connection with the web server, bypassing the external proxy server.
4. The SRX Series Firewall sends the connect response from the web server to the client.
5. For the remaining traffic, the SRX Series Firewall operates in pass-through mode and allows the HTTP connect request to go to the external proxy server.

Example—Configure Secure Web Proxy on an SRX Series Firewall

IN THIS SECTION

This example shows how to configure secure Web proxy on SRX Series Firewalls.

Hardware and Software Requirements

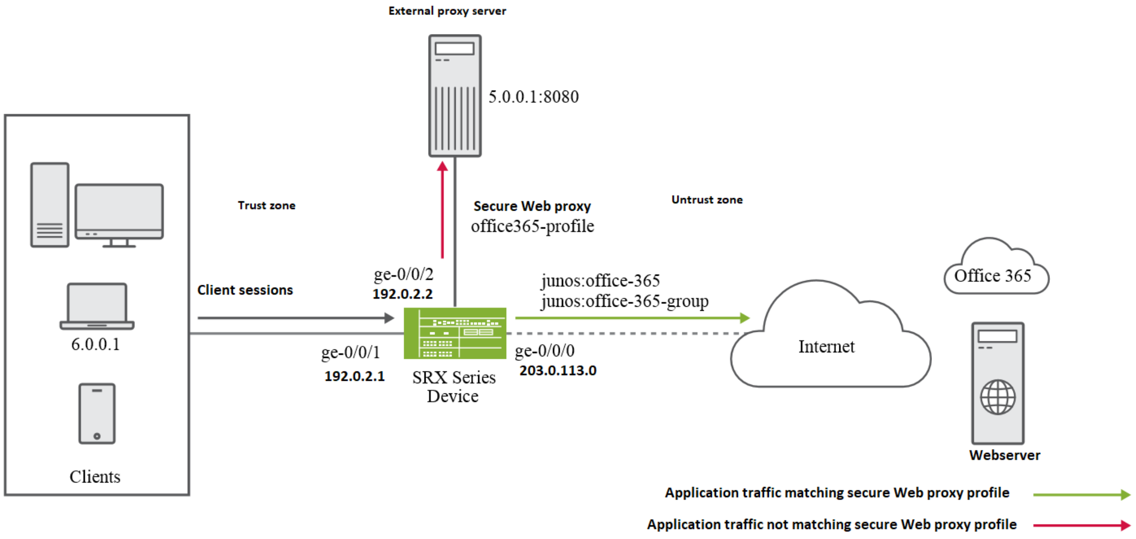
This example uses the following hardware and software components:

- A Juniper Networks SRX Series Firewall (SRX300, SRX320, SRX340, SRX345, SRX550, SRX1500, SRX4100, SRX4200, or vSRX Virtual Firewall).
- Junos OS Release 19.2R1 or later. We've tested this example using Junos OS Release 19.2R1.
- IP address and port number of the external proxy server.

Topology

No Link Title shows the topology used in this example.

Figure 6: Topology For Configuring Secure Web Proxy



In this example, the interfaces ge-0/0/1 and ge-0/0/2 are in the trust zone and are connected to the client and external proxy server, respectively. The interface ge-0/0/0 is in the untrust zone and is connected to the webserver through the Internet gateway. You configure a secure Web proxy profile, specifying Office 365 applications and external proxy details.

After you complete the configuration, the SRX Series Firewall will forward the Office 365 traffic directly to the webserver, bypassing the external proxy server for Office 365 traffic.

Configuration

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 203.0.113.0
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.1
set interfaces ge-0/0/2 unit 0 family inet address 192.0.2.2
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic system-
services all
set security zones security-zone trust interfaces ge-0/0/1.0 host-inbound-traffic system-
services all
set security zones security-zone trust interfaces ge-0/0/2.0 host-inbound-traffic system-
services all
set services application-identification application-group office-365-group applications
junos:OUTLOOK
set services application-identification application-group office-365-group applications
junos:OFFICE365-CREATE-CONVERSATION
set services web-proxy secure-proxy profile office365-profile proxy-address external_proxy ip
5.0.0.1/32
set services web-proxy secure-proxy profile office365-profile proxy-address external_proxy port
8080
set services web-proxy secure-proxy profile office365-profile dynamic-web-application
junos:office-365
set services web-proxy secure-proxy profile office365-profile dynamic-web-application-group
office-365-group
set security policies from-zone trust to-zone untrust policy 1 match source-address any
set security policies from-zone trust to-zone untrust policy 1 match destination-address any
set security policies from-zone trust to-zone untrust policy 1 match application any
set security policies from-zone trust to-zone untrust policy 1 then permit application-services
web-proxy profile-name office365-profile
```

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.

In this procedure you configure interfaces and security zones.

1. Configure the interfaces.

```
[edit]
user@host# set interfaces ge-0/0/0 unit 0 family inet address 203.0.113.0
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.1
user@host# set interfaces ge-0/0/2 unit 0 family inet address 192.0.2.2
```

2. Assign the interfaces to the security zones and configure the inbound traffic for all system services.

```
[edit]
user@host#  set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-
traffic system-services all
user@host#  set security zones security-zone trust interfaces ge-0/0/1.0 host-inbound-traffic
system-services all
user@host# set security zones security-zone trust interfaces ge-0/0/2.0 host-inbound-traffic
system-services all
```

3. Configure a custom application group for Office 365.

```
[edit]
user@host# set services application-identification application-group office-365-group
applications junos:OUTLOOK
user@host# set services application-identification application-group office-365-group
applications junos:OFFICE365-CREATE-CONVERSATION
```

4. Create a security proxy profile by specifying the Office 365 application details and the IP address and port details of the external proxy server.

```
[edit]
user@host#  set services web-proxy secure-proxy profile office365-profile proxy-address
external_proxy ip 5.0.0.1/32
user@host#  set services web-proxy secure-proxy profile office365-profile proxy-address
external_proxy port 8080
user@host#  set services web-proxy secure-proxy profile office365-profile dynamic-web-
application junos:office-365
user@host#  set services web-proxy secure-proxy profile office365-profile dynamic-web-
application-group office-365-group
```

5. Define the security policy for the traffic originating from the client to the Internet gateway device.

```
[edit]
user@host#  set security policies from-zone trust to-zone untrust policy 1 match source-
address any
user@host#  set security policies from-zone trust to-zone untrust policy 1 match destination-
address any
user@host#  set security policies from-zone trust to-zone untrust policy 1 match application
any
```

6. Define the policy action to apply the secure Web proxy profile on the permitted traffic.

```
[edit]
user@host#  set security policies from-zone trust to-zone untrust policy 1 then permit
application-services web-proxy profile-name office365-profile
```

The SRX Series Firewall forwards the Office 365 application traffic directly to the Office 365 server, bypassing the external proxy server. Other sessions that do not match the Office 365 application are routed to the external proxy server.

Results

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

```
[edit ]
user@host# show services web-proxy secure-proxy
profile office365-profile {
    proxy-address external_proxy {
        ip 5.0.0.1/32;
        port 8080;
    }
    dynamic-web-application junos:office-365
    dynamic-web-application-group office-365-group
}
```

```
[edit]
user@host# show security policies
from-zone trust to-zone untrust {
    policy 1 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit {
                application-services {
                    web-proxy {
                        profile-name office365-profile;
                    }
                }
            }
        }
    }
}
```

```
[edit]
    user@host# show security zones
    security-zone untrust {
    interfaces {
        ge-0/0/0.0 {
            host-inbound-traffic {
                system-services {
                    all;
                }
            }
        }
    }
}
security-zone trust {
    interfaces {
        ge-0/0/1.0 {
            host-inbound-traffic {
                system-services {
                    all;
                }
            }
        }
        ge-0/0/2.0 {
            host-inbound-traffic {
                system-services {
                    all;
                }
            }
        }
    }
}
```

Verification

IN THIS SECTION

Verify Session Details

Purpose

Verify the details of the session in which the secure Web proxy is applied.

Action

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

```
Session ID: 477, Policy name: 1/5, Timeout: 1796, Valid
          In: 6.0.0.1/63638 --> 5.0.0.1/8080;tcp, Conn Tag: 0x0, If: ge-0/0/0.0, Pkts: 22,
Bytes: 2451,
          Out: 5.0.0.1/8080 --> 6.0.0.1/63638;tcp, Conn Tag: 0x0, If: ge-0/0/1.0, Pkts: 0,
Bytes: 0,
        Session ID: 478, Policy name: 1/5, Timeout: 1796, Valid
          In: 6.0.0.1/63638 --> 13.107.7.190/443;tcp, Conn Tag: 0x0, If: ge-0/0/0.0, Pkts: 1, 
Bytes: 44, 
        Out: 13.107.7.190/443 --> 6.0.0.1/63638;tcp, Conn Tag: 0x0, If: ge-0/0/2.0, Pkts: 31, 
Bytes: 28898,
```

Meaning

In the sample output, the ID-477 is the client session and the ID-478 is the proxy session. In the second session, notice that the traffic from client 6.0.0.1 is directly going to the webserver 13.107.7.190.

Display Secure Web Proxy Session Statistics

Purpose

Display the details of the session in which the secure Web proxy is applied.

Action

From operational mode, enter the `show services web-proxy session detail` and `show services web-proxy session summary` commands.

```
user@host> show services web-proxy session detail
Web Proxy sessions:
Client Session ID: 38569, Proxy Session ID: 38570
Client: 6.0.0.1/53454 ---> 5.0.0.1/8080
Proxy : 6.0.0.1/53454 ---> 13.107.7.190/443
Proxy Request: CONNECT:www.office.com:443
Dynamic Web App: junos:OFFICE365-CREATE-CONVERSATION

Client Session ID: 38562, Proxy Session ID: 38564
Client: 6.0.0.1/53451 ---> 5.0.0.1/8080
Proxy : 6.0.0.1/53451 ---> 40.126.5.35/443
Proxy Request: CONNECT:login.microsoftonline.com:443
Dynamic Web App: junos:OFFICE365-CREATE-CONVERSATION

Client Session ID: 38567, Proxy Session ID: 38568
Client: 6.0.0.1/53453 ---> 5.0.0.1/8080
Proxy : 6.0.0.1/53453 ---> 13.107.246.10/443
Proxy Request: CONNECT:aadcdn.msauth.net:443
```

```
Dynamic Web App: junos:OFFICE365-CREATE-CONVERSATION

Client Session ID: 38571, Proxy Session ID: 0
Client: 6.0.0.1/53455 ---> 5.0.0.1/8080
Proxy : 6.0.0.1/53455 ---> 52.96.40.242/443
Proxy Request: CONNECT:outlook.office365.com:443
Dynamic Web App: junos:OWA

Client Session ID: 38561, Proxy Session ID: 38565
Client: 6.0.0.1/53450 ---> 5.0.0.1/8080
Proxy : 6.0.0.1/53450 ---> 40.126.5.35/443
Proxy Request: CONNECT:login.microsoftonline.com:443
Dynamic Web App: junos:OFFICE365-CREATE-CONVERSATION
```

```
user@host> show services web-proxy session summary

Web Proxy sessions:
Client Session                                                                  Proxy Session
[477] 6.0.0.1/63638 ---> 5.0.0.1/8080                             [478] 6.0.0.1/63638 --->
13.107.7.190/443
```

Meaning

In these samples, notice the details of the client session and the proxy session. You can also see proxy requests and dynamic web applications.

3

CHAPTER

Application Services Modules

Application Firewall

IN THIS SECTION

Application firewall (AppFW) provides policy-based enforcement and control on traffic based on application signatures. By using AppFW, you can block any application traffic not sanctioned by the enterprise. For more information, see the following topics:

Application Firewall Overview

IN THIS SECTION

This topic includes the following sections:

Limitations with Stateful Firewalls

Traditionally stateful firewalls used to control applications such as HTTP, SMTP, and DNS because these applications used well-known standards ports only. However, now it is possible to run these applications on any port as long as the client and server are using same protocol and same ports. Because of this standard stateful firewalls are not able to detect evasive applications. Additionally, with the growing popularity of Web applications and the shift from traditional full client-based applications to the Web, more and more traffic is being transmitted over HTTP.

This limitation of stateful firewalls, in which firewalls inspect traffic based on Layer 3 and Layer 4, left open to allow application layer exploits.

Application Firewall

Juniper Networks' application firewall (AppFW) leverages the results from the application identification to make an informed decision to permit, deny, reject, or redirect the traffic based on applications. AppFW enables you to enforce the policy control on Layer 7 traffic.

A predefined signature database is available on the Juniper Networks Security Engineering website. This database includes a library of application signatures. See Application Signatures for more details. These signature pages will give you visibility into the application category, group, risk-level, ports, and so on.

The AppFW allows you to block the applications based on their application signatures, while still allowing other HTTP traffic to pass through the firewall. For example, an application firewall rule could block HTTP traffic from Facebook but allow Web access to HTTP traffic from MS Outlook.

Benefit of Application Firewall

- Provides granular security control to high-risk applications based on user-defined policies.
- Adds flexibility by providing policy control over application access based on the requirements.

Application Firewall with Unified Policies

Starting in Junos OS release 18.2R1, you can use unified policies to avail the same functionality of an AppFW configuration. Unified policies leverage the application identity information from the application identification (AppID) service to permit, deny, reject, or redirect the traffic. A unified policy configuration handles all application firewall functionality and simplifies the task of configuring a firewall policy.

Read one of the following topic for configuring AppFW:

- If you are using Junos OS version 18.2 and later releases, you must configure Unified policies to get same benefits as traditional AppFW. See "Application Firewall Support with Unified Policies" on page 132.

- If you are using Junos OS version prior to Junos OS 18.2, you can configure traditional AppFW. See "Application Firewall Overview" on page 130.

Application Firewall Support with Unified Policies

Starting in Junos OS Release 18.2R1, SRX Series Firewalls and vSRX Virtual Firewall instances support unified policies, allowing granular control and enforcement of Layer 7 dynamic applications within the traditional security policy.

Unified policies are the security policies that enable you to use dynamic applications as match conditions as part of the existing 5-tuple or 6-tuple (5-tuple with user firewall) match conditions to detect application changes over time.

- If you are planning to upgrade to Junos OS Release 18.2R1 and later releases, note the following points regarding using APPFW functionality:
  - All existing AppFW related CLI statements and commands are deprecated. That is—

    Starting in Junos OS Release 18.2R1 Application Firewall (AppFW) functionality is deprecated—rather than immediately removed—to provide backward compatibility and an opportunity to bring your configuration into compliance with the new configuration. As a part of this change, the `[edit security application-firewall]` hierarchy and all the configuration options under this hierarchy are deprecated.
  - AppFW functionality works if you continue to configure in the deprecated hierarchy. You can configure AppFW in the deprecated hierarchy in CLI by manual input only.
  - Configuring a traditional AppFW policy and a unified policy in the same security policy is not supported. The system displays the following error message if you attempt to do so:

    ```
    Traditional AppFW and dynamic-application can't be applied to same policy
    ```
- If you are downgrading from Junos OS Release 18.2R1 to any earlier versions of Junos OS:
  - You must delete all unified policies to avoid a commit check failure after a downgrade.

For example on configuring a unified policies, see Configuring Unified Security Policies.

SEE ALSO

Application Identification Support for Unified Policies | 95

Example: Configure Application Firewall with Unified Policy

IN THIS SECTION

This example describes how to configure a unified policy to allow or block traffic based on the applications.

System Requirements

System Requirements

This example uses the following hardware and software components:

- SRX Series Firewall running Junos OS Release 18.2R1. This configuration example is tested with Junos OS release 19.1R1.

Before You Begin

- Install a valid application identification feature license on your SRX Series Firewall. See Managing Junos OS Licenses.

- Download and install the Junos OS application signature package. Downloading and Installing the Junos OS Application Signature Package.

Overview

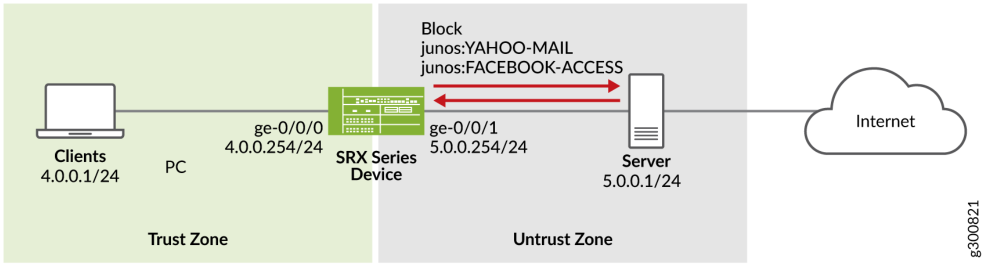
IN THIS SECTION

In this example, you create a very common scenario to block certain application and application group such as Yahoo-Mail and Facebook-Access.

Topology

This example uses the topology as shown in Figure 7 on page 134.

Figure 7: Topology For Unified Policies Example



This example uses following zones and interfaces configuration.

- The client system is connected to the ge-0/0/0.0 interface with IP address 4.0.0.254/24. It is part of the trust zone.
- The server system is connected to the ge-0/0/1.0 interface with IP address 5.0.0.254/24. It is part of the untrust zone.

Create a security policy configuration to block certain applications using the following steps:

- Create a security policy for the traffic from zone trust to untrust to block the access to the Yahoo-Mail or Facebook-Access applications.
- Create a redirect message for the denied or rejected traffic to inform the user about the status of their request.
- Create a default policy to allow rest of the traffic.

Configuration

IN THIS SECTION

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 dynamic-application profile profile1 redirect-message type custom-text content
"THIS APPLICATION IS BLOCKED"
set security policies from-zone trust to-zone untrust policy policy-1 match source-address any
set security policies from-zone trust to-zone untrust policy policy-1 match destination-address
any
set security policies from-zone trust to-zone untrust policy policy-1 match application any
set security policies from-zone trust to-zone untrust policy policy-1 match dynamic-application
junos:YAHOO-MAIL
set security policies from-zone trust to-zone untrust policy policy-1 match dynamic-application
junos:FACEBOOK-ACCESS
set security policies from-zone trust to-zone untrust policy policy-1 then reject profile
profile1
set security policies default-policy permit-all
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust interfaces ge-0/0/1.0
set interfaces ge-0/0/0 unit 0 family inet address 4.0.0.254/24
set interfaces ge-0/0/1 unit 0 family inet address 5.0.0.254/24
```

Procedure

Step-by-Step Procedure

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 unified policy using dynamic applications:

1. Configure security zones and interfaces.

```
[edit]
user@host#set security zones security-zone trust host-inbound-traffic system-services all
user@host#set security zones security-zone trust interfaces ge-0/0/0.0
user@host#set security zones security-zone untrust host-inbound-traffic system-services all
user@host#set security zones security-zone untrust interfaces ge-0/0/1.0
user@host#set interfaces ge-0/0/0 unit 0 family inet address 4.0.0.254/24
user@host#set interfaces ge-0/0/1 unit 0 family inet address 5.0.0.254/24
```

2. Create redirect profile.

```
[edit]
user@host#set security dynamic-application profile profile1 redirect-message type custom-text
content "THIS APPLICATION IS BLOCKED"
```

3. Create a security policy with a dynamic application as the match criteria.

```
[edit]
user@host#set security policies from-zone trust to-zone untrust policy policy-1 match source-
address any
user@host#set security policies from-zone trust to-zone untrust policy policy-1 match
destination-address any
user@host#set security policies from-zone trust to-zone untrust policy policy-1 match
application any
user@host#set security policies from-zone trust to-zone untrust policy policy-1 match dynamic-
application junos:YAHOO-MAIL
user@host#set security policies from-zone trust to-zone untrust policy policy-1 match dynamic-
application junos:FACEBOOK-ACCESS
user@host#set security policies from-zone trust to-zone untrust policy policy-1 then reject
profile profile1
```

4. Create a default policy to permit the remaining traffic.

```
[edit]
user@host#set security policies default-policy permit-all
```

Results

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

```
[edit]
user@host# show security
dynamic-application {
    profile profile1 {
        redirect-message {
            type {
                custom-text {
                    content "THIS APPLICATION IS BLOCKED";
                }
            }
        }
    }
}
policies {
    from-zone trust to-zone untrust {
        policy policy-1 {
            match {
                source-address any;
                destination-address any;
                application any;
                dynamic-application [junos:YAHOO-MAIL junos:FACEBOOK-ACCESS ];
            }
            then {
                reject {
                    profile profile1;
                }
            }
        }
    }
    default-policy {
        permit-all;
    }
}
zones {
    security-zone trust {
        host-inbound-traffic {
            system-services {
                ping;
            }
        }
        interfaces {
            ge-0/0/0.0;
        }
    }
    security-zone untrust {
        host-inbound-traffic {
            system-services {
                ping;
            }
        }
        interfaces {
            ge-0/0/1.0;
        }
    }
}
```

```
[edit]
user@host# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 4.0.0.254/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 5.0.0.254/24;
        }
    }
}
fxp0 {
    unit 0 {
        family inet {
            address 10.102.70.185/24;
        }
    }
}
```

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

Verification

Use the following procedures to verify if the policy configuration.

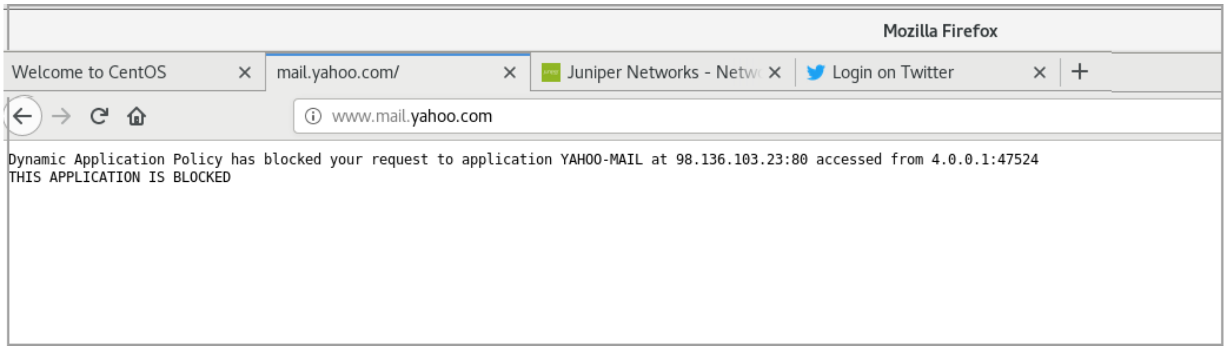
Verifying Policy Action

Purpose

Verify that the unified policy has blocked that configured applications.

Action

From your Web browser, try to access the application. For example, Yahoo-Mail. The system displays the redirect message as shown in the following image.



```
Dynamic Application Policy has blocked your request to application YAHOO-MAIL at 98.136.103.23:80 accessed from 4.0.0.1:47524
THIS APPLICATION IS BLOCKED
```

Meaning

Whenever the security policy rejects traffic based on the dynamic application, the output displays the redirect message as configured by you in the dynamic application profile.

Verifying Unified Policy Configuration

Purpose

Verify that the unified policy configuration is correct.

Action

From operational mode, enter the `show security policies detail` command to display a detailed summary of all security policies on the device.

```
user@host> show security policies detail

Default policy: permit-all
Pre ID default policy: permit-all
Policy: policy-1, action-type: reject, State: enabled, Index: 7, Scope Policy: 0
  Policy Type: Configured
  Sequence number: 1
  From zone: trust, To zone: untrust
  Source vrf group:
    any
  Destination vrf group:
    any
  Source addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Destination addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination ports: [0-0]
  Dynamic Application:
junos:FACEBOOK-ACCESS: 244
junos:YAHOO-MAIL: 236
dynapp-redir-profile: profile1(1)
  Per policy TCP Options: SYN check: No, SEQ check: No, Window scale: No
```

Meaning

The output displays information about security policy. Verify the following information:

- Configured policy name policy-1 and policy action reject.
- Configured dynamic applications junos:FACEBOOK-ACCESS and junos:YAHOO-MAIL.
- Redirect profile profile1.

SEE ALSO

> *dynamic-application (Security Policies)*

Traditional Application Firewall

IN THIS SECTION

This topic includes the following sections:

Understanding How Application Firewall Works

As you can use existing security policy to enforce traditional firewall controls on the traffic, you can use AppFW module to block certain application traffic, based on their application signatures, while still allowing other HTTP traffic to pass through the firewall.

Security device processes traffic in the following sequence when you have configured a AppFW:

1. Security policy matches the zone pair specified in the policy.

2. Security policy matches the packets with matching conditions (source and destination IP addresses, source and destination ports, and application type)

3. Security policy applies one of the following actions to the matching traffic.

   - Reject—Notify the client, drop the traffic, and log the event.

   - Deny—Drop the traffic, and log the event.

   - Permit—Open a session, log the event, and apply services as specified.

     - Invoke application services to retrieve the application ID for the traffic.

     - Apply the specified application firewall rule set.

   NOTE: If you are using Junos OS Release 20.1 or later releases and have configured HTTP-based custom application signature, the legacy application firewall redirect action might not work for HTTPS traffic. Instead of redirecting the HTTPS traffic, the security device denies or rejects the traffic.

   NOTE: All IP fragmented packets received on the security device must be reassembled before forwarding.

Application Firewall Rule Sets and Rules

Consider following when configuring application firewall:

- You can apply one AppFW rule set to multiple different security policies.

- You can configure an AppFW inside a logical system.

- You can configure multiple dynamic applications in a rule and multiple rules in a rule set. However, there is a limit to the overall number of rule sets and rules.

- You can configure a dynamic application group as match criteria in a rule. An application group includes multiple related applications. For more information, see "Predefined and Custom Application Groups for Application Identification" on page 87.

- The default rule defines the action required for any traffic that does not match any rule. So, a AppFW rule set must contain a default rule.

Application Firewall with ALG

On your security devices, when you enable ALG, application identification includes the ALG results to identify the applications in the control session. AppFW permits ALG data sessions whenever control sessions are permitted. If the control session is denied, there will be no data sessions. If you disable ALG, application identification relies on signatures to identify the application in the control and data sessions. If a signature match is not found, the application is considered unknown. AppFW handles the applications based on the application identification result.

Unknown Applications

Application identification classifies unknown dynamic applications with ID junos:UNKNOWN. AppID uses the reserved keyword junos:UNKNOWN in the following cases

- The traffic does not match an application signature in the database.
- The system encounters an error when identifying the application.
- The session fails over to another device.

Traffic with an application ID of junos:UNKNOWN matches a rule with a dynamic application of junos:UNKNOWN. If there is no rule defined for junos:UNKNOWN, the default rule is applied.

Session Logging for Application Firewalls

You can log the traffic by enabling the log option under a security policy. Note the following while you inspect a log message when AppFW is configured as given in Table 5 on page 143:

Table 5: Session Logging for Application Firewall Configuration

Security Policy Action	Log Creation	More Details
Permit	Creates a session and logs a session create message	When security policy permit action creates a session even before the AppFW rules are applied, log message includes one of the following update: • If the application is already identified, it’s information is added to the session create message. • If the application is in the process of being identified, the dynamic application field are updated as UNKNOWN.

Table 5: Session Logging for Application Firewall Configuration *(Continued)*

Security Policy Action	Log Creation	More Details
Reject/Deny	Logs reject or deny message, but does not create a session.	When a AppFW rule denies or rejects traffic, the log message includes one of the following phrases in the reason field: • `appfw deny` or `appfw deny redirect` • `appfw reject` or `appfw reject redirect` • `policy deny` • `policy reject`

Application Firewall Support in Chassis Cluster

When your security device is in chassis cluster mode, the AppFW action before and after the failover depends on the application identification state, as shown in Table 6 on page 144.

Table 6: Application Firewall Actions

Before Failover		After Failover	
Application ID State	**Application Firewall Action**	**Application ID State**	**Application Firewall Action**
Success	Deny	Success	Deny
Success	Permit	Success	Permit
Pending	—	UNKNOWN	Action based on the rule defined for unknown application If there is no rule defined for unknown, then the default rule is applied

Note the following when you have your security device in chassis cluster mode:

- When you enable application identification, the pre-match state application IDs are not synced to other node. If there are any failover sessions, which were still under classification, will not have any application IDs assigned. This could result in application statistics and counters mismatch.
- In-service software upgrade (unified ISSU) is not supported due to lack of *chassis cluster* infrastructure support. Thus, the failover event is controlled through the application firewall policy by allowing or denying the unknown dynamic applications.

SEE ALSO

Understanding Security Policy Elements

Security Policies Overview

Understanding Security Policy Rules

Creating Redirects in Application Firewall

IN THIS SECTION

When AppFW denies or rejects traffic, it does not notify clients that such action is taken. Clients being unaware that their request is rejected, might keep on trying to access the Web page. To alleviate this inconvenience, the Junos OS allows you to provide an explanation for the action or to redirect the client to an informative webpage. Following examples show you how to create a redirect message.

Redirect with Block Message

Use the `block-message` option with the `reject` or `deny` action in AppFW rule.

```
.....
    rule 1 {
    match {
        dynamic-application junos:FACEBOOK-CHAT
    }
    then {
        reject {
            block-message;
        }
    }
    }
.....
```

When AppFW rejects the traffic, a splash screen displays the following default message to the user:

```
user-name, Application Firewall has blocked your request to application FACEBOOK-CHAT at dst-
ip:dst-port accessed from src-ip:src-port.
```

Customize Redirect Message

You can customize the redirect action by including additional text on the splash screen or by specifying a URL to which you can redirect a user. To customize the block message, you must create a block message profile at `[edit security application-firewall]` hierarchy level and define the type and content as shown in the following sample.

```
...
    profile Redirect-Profile {
        block-message {
            type {
                custom-text {
                    content "YOUR APPLICATION IS BLOCKED AS PER THE ORGANIZATION POLICY";
                }
            }
        }
    }
...
```

Next, you refer the block message profile in the AppFW rule set, and apply it to one or more of the rules using the `block-message` option;

```
rule-sets Ruleset-1 {
    rule 1 {
        match {
            dynamic-application junos:FACEBOOK-CHAT;
        }
        then {
            reject {
                block-message;
            }
        }
    }
    profile Redirect-Profile;
}
```

In this case, AppFW displays the configured block message whenever it rejects the traffic based on the configured rule.

Customize Redirect Message with URL

When AppFW rejects or redirects the traffic, you can redirect the client to the specified Web page for further action. The URL can be hosted on either the SRX Series Firewall or an external server.

You can set the redirects to the other server by configuring block-message type as custom-redirect-url as shown in the sample below:

```
profile Redirect-Profile {
    block-message {
        type {
            custom-redirect-url {
                content http://abc.company.com/information;
            }
        }
    }
}
```

Next, you refer the block message profile in the AppFW rule set, and apply to one or more of the rules using the `block-message` option as shown in the following sample:

```
rule-sets Ruleset-1 {
    rule 1 {
        match {
            dynamic-application junos:FACEBOOK-CHAT;
        }
        then {
            reject {
                block-message;
            }
        }
    }
    profile Redirect-Profile;
}
```

In this case, AppFW redirects the use to the URL http://abc.company.com/information whenever it rejects the traffic based on the configured rule.

Example: Configuring Application Firewall

IN THIS SECTION

This example shows how to configure application firewall rule sets within the security policy.

Before You Begin

- Valid application identification feature license installed on an SRX Series Firewall. See Managing Junos OS Licenses.

- Download and install the Junos OS application signature package. Downloading and Installing the Junos OS Application Signature Package.

System Requirements

- SRX Series Firewall with Junos OS Release 15.1X49-D60 or later. This configuration example is tested for Junos OS Release 15.1X49-D60.

Overview

In this example, you create application firewall for the following two common scenarios as described in Table 7 on page 149.

Table 7: Configure Application Firewall to Permit or Deny Traffic

Objectives	Steps to Follows	Results
Block a certain application and allow other applications	Configure a security policy to allow HTTP traffic.	Security policy permits or drops the traffic based on matching specified Layer 3 or Layer 4 criteria.
	Configure an AppFW rule set with following options: • Rules with dynamic applications that you want to block • Action to deny dynamic application traffic. • Default rule to permit other traffic	AppFW assess the permitted traffic at Layer 7 based on its application ID.
	Refer the AppFW rule set in the security policy.	• AppFW blocks the traffic matching the configured dynamic applications. • Default policy permits other traffic.
Allow a certain application and block other applications	Configure a security policy to allow HTTP traffic.	Security policy permits or drops the traffic based on matching specified Layer 3 or Layer 4 criteria.
	Configure an AppFW rule set with following options: • Rules with dynamic applications that you want to permit • Action to permit dynamic application traffic. • Default rule to block other traffic.	AppFW assess the permitted traffic at Layer 7 based on its application ID.

Table 7: Configure Application Firewall to Permit or Deny Traffic *(Continued)*

Objectives	Steps to Follows	Results
	Refer the AppFW rule set in the security policy.	• AppFW permits the traffic matching the configured dynamic applications. • Default policy blocks other traffic.

NOTE: On all SRX Series Firewalls, J-Web pages for AppSecure Services are preliminary. We recommend using CLI for configuration of AppSecure features.

Configuration

IN THIS SECTION

Application Firewall Rule to Explicitly Deny Certain Application and Permit All Else

In this example, you block dynamic-applications junos:FACEBOOK-CHAT junos:FACEBOOK-FARMVILLE and allow remaining traffic.

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 policies from-zone untrust to-zone trust policy policy1 match source-address any
set security policies from-zone untrust to-zone trust policy policy1 match destination-address
any
set security policies from-zone untrust to-zone trust policy policy1 match application junos-
http
```

```
set security policies from-zone untrust to-zone trust policy policy1 then permit application-
services application-firewall rule-set rs1
set security application-firewall rule-sets rs1 rule r1 match dynamic-application
[junos:FACEBOOK-CHAT,junos:FACEBOOK-FARMVILLE ]
set security application-firewall rule-sets rs1 rule r1 then deny
set security application-firewall rule-sets rs1 default-rule 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 *CLI User Guide*.

To configure two security policies with application firewall rule sets that permit or deny traffic from different dynamic applications:

1. Define the application firewall rule set to deny traffic from selected dynamic applications.

   ```
   [edit security application-firewall rule-sets rs1]
   user@host# set rule r1 match dynamic-application [junos:FACEBOOK-CHAT,junos:FACEBOOK-
   FARMVILLE]
   user@host# set rule r1 then deny
   user@host# set default-rule permit
   ```

2. Configure the security policy to allow HTTP traffic and invoke application firewall rule set rs1.

   ```
   [edit security policies from-zone untrust to-zone trust policy policy1]
   user@host# set match source-address any
   user@host# set match destination-address any
   user@host# set match application junos-http
   user@host# set then permit application-services application-firewall rule-set rs1
   ```

Results

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

```
[edit]
user@host# show security policies
```

```
    from-zone untrust to-zone trust {
        policy 1 {
            match {
                source-address any;
                destination-address any;
                application junos-http;
            }
            then {
                permit {
                    application-services {
                        application-firewall {
                            rule-set rs1;
                        }
                    }
                }
            }
        }
    }
user@host# show security application-firewall
    rule-sets rs1 {
        rule r1 {
            match {
                dynamic-application  [junos:FACEBOOK-CHAT,junos:FACEBOOK-FARMVILLE];
            }
            then {
                 deny;
            }
        }
        default-rule {
            permit;
        }
    }
```

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

Application Firewall Rule to Explicitly Permit Certain Application and Deny All Else

In this example, you permit dynamic-applications junos:FACEBOOK-ACCESS and block remaining traffic.

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 policies from-zone untrust to-zone trust policy policy2 match source-address any
set security policies from-zone untrust to-zone trust policy policy2 match destination-address
any
set security policies from-zone untrust to-zone trust policy policy2 match application any
set security policies from-zone untrust to-zone trust policy policy2 then permit application-
services application-firewall rule-set rs2
set security application-firewall rule-sets rs2 rule r1 match dynamic-application
[junos:FACEBOOK-ACCESS junos:UNKNOWN]
set security application-firewall rule-sets rs2 rule r1 then permit
set security application-firewall rule-sets rs2 default-rule deny
```

Step-by-Step Procedure

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

To configure two security policies with application firewall rule sets that permit or deny traffic from different dynamic applications:

1. Configure a security policy to process any traffic that does not go to the HTTP static ports with the application firewall rule set rs2.

```
[edit security policies from-zone untrust to-zone trust policy policy2]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application junos:http
user@host# set then permit application-services application-firewall rule-set rs2
```

2. Define the application firewall rule set to permit traffic from selected dynamic applications.

```
[edit security application-firewall rule-sets rs2]
user@host# set rule r1 match dynamic-application [junos:FACEBOOK-ACCESS, junos:UNKNOWN]
user@host# set rule r1 then permit
user@host# set default-rule deny
```

Results

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

```
[edit]
user@host# show security policies
    from-zone untrust to-zone trust {
        policy 2 {
            match {
                source-address any;
                destination-address any;
                application junos:http;
            }
            then {
                permit {
                    application-services {
                        application-firewall {
                            rule-set rs2;
                        }
                    }
                }
            }
        }
    }
user@host# show security application-firewall
    rule-sets rs2 {
        rule r1 {
            match {
                dynamic-application [junos:FACEBOOK-ACCESS, junos:UNKNOWN];
            }
            then {
                 permit;
            }
        }
        default-rule {
            deny;
        }
    }
```

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

Verification

IN THIS SECTION

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

Verifying Application Firewall Configuration

Purpose

Verify information about application firewall support enabled under the security policy.

Action

To verify the security policy configuration enabled with application firewall, enter the `show security policies` and `show security policies detail` commands. To verify all the application firewall rule sets configured on the device, enter the `show security application-firewall rule-set all` command.

Meaning

The output displays information about application firewall enabled policies configured on the system. Verify the following information.

- Rule set
- Rules
- Match criteria

SEE ALSO

> *Security Policies Configuration Overview*
>
> *Example: Configuring a Security Policy to Permit or Deny All Traffic*

Example: Configuring Application Firewall with Application Groups

IN THIS SECTION

The application identification (AppID) module manages predefined application groups. An application group includes related applications under a single name for simplified, consistent reuse when using in any application services. An application group can contain multiple applications and application groups simultaneously. It is possible to assign one application to multiple groups.

You can configure a AppFW rule to permit or to deny traffic by specifying a predefined application group along with applications as match criteria.

Advantage of using predefined application groups is - As the application signature database changes, the predefined application group is modified automatically to include new signatures. In this case, if you already have a AppFW rule with predefined application group, the inclusion of new signatures in the application group does not affect the existing AppFW rule.

This example shows how to configure application groups in a AppFW rule set.

Before You Begin

- Install a valid application identification feature license on your SRX Series Firewall. See Managing Junos OS Licenses.
- Download and install the Junos OS application signature package. Downloading and Installing the Junos OS Application Signature Package.

System Requirements

- SRX Series Firewall with Junos OS Release 15.1X49-D60 or later. This configuration example is tested for Junos OS Release 15.1X49-D60.

Overview

In this example, you configure a security policy to control outbound traffic from the trust zone to the untrust zone. Next you create a AppFW rule to allow specific application traffic (junos:GOOGLETALK), but deny all other known similar application traffic (social networking traffic) using application group.

It is very important to note the order of AppFW rules because, the predefined group junos:social-networking includes the junos:GOOGLETALK application. To allow junos:GOOGLETALK traffic and deny the rest of the group, you must place the rule permitting junos:GOOGLETALK traffic before the rule denying traffic from the rest of the applications in the group.

Configuration

IN THIS SECTION

Procedure

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 application-firewall rule-sets social-network rule google-rule match dynamic-
application junos:GOOGLETALK
set security application-firewall rule-sets social-network rule google-rule then permit
set security application-firewall rule-sets social-network rule denied-sites match dynamic-
application-groups junos:social-networking
 set security application-firewall rule-sets social-network rule denied-sites match dynamic-
application junos:UNKNOWN
set security application-firewall rule-sets social-network rule denied-sites then deny
set security application-firewall rule-sets social-network default-rule permit
set security policies from-zone trust to-zone untrust policy outbound-traffic match source-
address any
set security policies from-zone trust to-zone untrust policy outbound-traffic match destination-
address any
set security policies from-zone trust to-zone untrust policy outbound-traffic match application
junos:HTTP
```

```
set security policies from-zone trust to-zone untrust policy outbound-traffic then permit
application-services application-firewall rule-set social-network
```

Step-by-Step Procedure

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

To configure application firewall rule-sets and security policies for outbound traffic:

1. Create the rule-set social-network.

   ```
   [edit]
   user@host# set security application-firewall rule-sets social-network
   ```

2. Define a rule to permit Google-Talk traffic.

   ```
   [edit security application-firewall rule-sets social-network]
   user@host# set rule google-rule match dynamic-application junos:GOOGLETALK
   user@host# set rule google-rule then permit
   ```

3. Define a second rule that denies all other social-networking traffic and traffic from an unknown application.

   ```
   [edit security application-firewall rule-sets social-network]
   user@host# set rule denied-sites match dynamic-application-groups junos:social-networking
   user@host# set rule denied-sites match dynamic-application junos:UNKNOWN
   user@host# set rule denied-sites then deny
   ```

   Note that the rule sequence is very important. You must place the rule with junos:GOOGLETALK before the rule with junos:social-networking. Otherwise, AppFW rule denies even GOOGLETALK traffic along junos:social-networking.

4. Define the default-rule that permits all other traffic.

   ```
   [edit security application-firewall rule-sets social-network]
   user@host# user@host# set default-rule permit
   ```

5. Configure the outbound-traffic policy to apply the social-network rule-set to all outbound traffic.

```
[edit security policies from-zone trust to-zone untrust policy outbound-traffic]
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application junos:HTTP
user@host# set then permit application-services application-firewall rule-set social-network
```

Results

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

```
[edit]
user@host# show security application-firewall
...
rule-sets social-network {
    rule google-rule {
        match {
            dynamic-application junos:GOOGLETALK;
        }
    }
        then {
            permit ;
        }
    rule denied-sites {
        match {
            dynamic-application-groups junos:social-networking
            dynamic-application junos:UNKNOWN;
        }
        then {
            deny ;
        }
    }
    default-rule {
        permit;
    }
}
...
```

```
[edit]
user@host# show security policies
from-zone untrust to-zone trust {
    ...
    policy outbound-traffic {
        match {
            source-address any;
            destination-address any;
            application junos-http;
        }
        then {
            permit {
                application-services {
                    application-firewall {
                        rule-set social-network
                    }
                }
            }
        }
    }
    ...
}
```

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

Verification

IN THIS SECTION

Verifying Application Firewall Configuration

Purpose

Verify information about application grouping support under the application firewall policy.

Action

- To verify the application firewall policy configuration enabled with application grouping, from the operational mode, enter the `show security policies` and `show security policies detail` commands.
- To verify all the application firewall rule sets configured on the device, from the operational mode, enter the `show security application-firewall rule-set all` command.
- To verify the list of applications defined within the application group, from the operational mode, enter the `show services application-identification application-group` *`application-group-name`* command.

SEE ALSO

Security Policies Configuration Overview

Customizing Application Groups for Junos OS Application Identification | 87

Example: Configuring Application Firewall When SSL Proxy Is Enabled

IN THIS SECTION

This example describes how to configure a AppFW when you have enabled the SSL proxy.

For **application junos-https**, SSL proxy detects an SSL session based on the dynamic application identified for that session. In case if any known Web servers are running nonstandard ports, you can use a custom Junos OS application to identify the application. However, if the Web servers are not known, for example on the Internet, you can use **application any**. Non-SSL sessions that come across the policy rule are ignored by SSL proxy. A syslog SSL_PROXY_SESSION_IGNORE is sent out for these sessions. Juniper Networks recommends that you use application “any” with caution because this can result in a lot of traffic, incurring initial SSL proxy processing and thereby impacting performance.

The security device bypasses SSL proxy services if when SSL proxy profile is attached to the security rule, when none of the services (AppFW, IDP, or AppTrack) are configured

Requirements

Before you begin:

- Install a valid application identification feature license on your SRX Series Firewall. See Managing Junos OS Licenses.
- Download and install the Junos OS application signature package. Downloading and Installing the Junos OS Application Signature Package.
- Create an application (or application set) that indicates that the policy applies to traffic of that type. See *Example: Configuring Security Policy Applications and Application Sets*.
- Create a SSL proxy profile that enables SSL proxy by means of a policy. See "Configuring SSL Forward Proxy" on page 387.

System Requirements

- SRX Series Firewall with Junos OS Release 15.1X49-D60 or later. This configuration example is tested for Junos OS Release 15.1X49-D60.

Overview

In this example, you configure two security policies with AppFW rule sets to permit or deny traffic from plain text or encrypted traffic:

- Allow the encrypted version of Oracle and deny any other encrypted traffic.
- Allow all HTTP traffic, except Hulu.

Configuration

IN THIS SECTION

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 policies from-zone Z_1 to-zone Z_2 policy policy1 match source-address any
set security policies from-zone Z_1 to-zone Z_2 policy policy1 match destination-address any
set security policies from-zone Z_1 to-zone Z_2 policy policy1 match application junos-https
set security policies from-zone Z_1 to-zone Z_2 policy policy1 then permit application-services
application-firewall rule-set appfw-rs-1
set security policies from-zone Z_1 to-zone Z_2 policy policy1 then permit application-services
ssl-proxy profile-name ssl-profile-1
set security policies from-zone Z_1 to-zone Z_2 policy policy2 match source-address any
set security policies from-zone Z_1 to-zone Z_2 policy policy2 match destination-address any
set security policies from-zone Z_1 to-zone Z_2 policy policy2 match application junos-http
set security policies from-zone Z_1 to-zone Z_2 policy policy2 then permit application-services
application-firewall rule-set appfw-rs-2
set security application-firewall rule-sets appfw-rs-1 rule rule1 match dynamic-application
[junos:ORACLE]
set security application-firewall rule-sets appfw-rs-1 rule rule1 then permit
set security application-firewall rule-sets appfw-rs-1 default-rule deny
set security application-firewall rule-sets appfw-rs-2 rule rule1 match dynamic-application
[junos:HULU]
set security application-firewall rule-sets appfw-rs-2 rule rule1 then deny
set security application-firewall rule-sets appfw-rs-2 default-rule permit
```

Procedure

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 *CLI User Guide*.

1. Configure a security policy to process the traffic with AppFW rule set and SSL proxy profile.

   ```
   [edit security policies from-zone Z_1 to-zone Z_2 policy policy1
   user@host# set match source-address any
   user@host# set match destination-address any
   user@host# set match application junos-https
   ```

```
user@host# set then permit application-services application-firewall rule-set appfw-rs-1
user@host# set then permit application-services ssl-proxy profile-name ssl-profile-1
```

2. Configure another security policy with AppFW rule set.

```
[edit security policies from-zone Z_1 to-zone Z_2 policy policy2
user@host# set match source-address any
user@host# set match destination-address any
user@host# set match application junos-http
user@host# set then permit application-services application-firewall rule-set appfw-rs-2
```

3. Define the AppFW rule set to permit an encrypted version of Oracle traffic and to deny any other encrypted traffic.

```
[edit security application-firewall rule-sets appfw-rs1]
user@host# set rule rule1 match dynamic-application [junos:ORACLE]
user@host# set rule rule1 then permit
user@host# set default-rule deny
```

4. Define another AppFW rule set to allow all plain text traffic except Hulu.

```
[edit security application-firewall rule-sets appfw-rs2]
user@host# set rule rule1 match dynamic-application [junos:HULU]
user@host# set rule rule1 then deny
user@host# set default-rule permit
```

Results

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

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

NOTE:

Verifying Application Firewall In an SSL Proxy Enabled Policy

Purpose

Verify that the application is configured correctly when SSL proxy is enabled in a policy.

Action

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

The following output shows the options for the `show security flow session` command.

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

```


    Possible completions:
     <[Enter]>            Execute this command
     application          Application protocol name
     application-firewall Show application-firewall sessions
     application-firewall-rule-set  Show application firewall sessions matching rule-set name
     brief                Show brief output (default)
     destination-port     Destination port (1..65535)
     destination-prefix   Destination IP prefix or address
     dynamic-application  Dynamic application name
     extensive            Show detailed output
+    encrypted            Show encrypted traffic
     family               Show session by family
     idp                  Show idp sessions
     interface            Name of incoming or outgoing interface
     nat                  Show sessions with network address translation
     protocol             IP protocol number
     resource-manager     Show sessions with resource manager
     session-identifier   Show session with specified session identifier
     source-port          Source port (1..65535)
     source-prefix        Source IP prefix or address
     summary              Show output summary
     tunnel               Show tunnel sessions
      |                    Pipe through a command
```

To display SSL encrypted UNKNOWN sessions, use the `show security flow session application-firewall dynamic-application junos:SSL extensive` command.

To display all HTTPS sessions, use the `show security flow session application-firewall dynamic-application junos:HTTP encrypted extensive` command.

SEE ALSO

SSL Proxy Overview | 350

Release History Table

Release	Description
18.2R1	Starting in Junos OS Release 18.2R1 Application Firewall (AppFW) functionality is deprecated— rather than immediately removed—to provide backward compatibility and an opportunity to bring your configuration into compliance with the new configuration. As a part of this change, the `[edit security application-firewall]` hierarchy and all the configuration options under this hierarchy are deprecated.

RELATED DOCUMENTATION

Application Identification | 5

Application Tracking | 166

Application QoS | 190

Advanced Policy-Based Routing | 220

SSL Proxy | 350

Application Tracking

IN THIS SECTION

Application tracking (AppTrack) is a logging and reporting tool that can be used to share information for application visibility. AppTrack sends log messages through syslog providing application activity update messages. For more information, see the following topics:

Understanding Application Tracking

IN THIS SECTION

AppTrack, an application tracking tool, provides statistics for analyzing bandwidth usage of your network. When enabled, AppTrack collects byte, packet, and duration statistics for application flows in the specified zone. By default, when each session closes, AppTrack generates a message that provides the byte and packet counts and duration of the session, and sends it to the host device. Juniper Secure Analytics (formally known as STRM) retrieves the data and provides flow-based application visibility.

AppTrack messages are similar to session logs and use syslog or structured syslog formats. The message also includes an application field for the session. If AppTrack identifies a custom-defined application and returns an appropriate name, the custom application name is included in the log message. (If the application identification process fails or has not yet completed when an update message is triggered, the message specifies `none` in the application field.)

AppTrack supports both IPv4 and IPv6 addressing. Related messages display addresses in the appropriate IPv4 or IPv6 format.

User identity details such as user name and user role have been added to the AppTrack session create, session close, and volume update logs. These fields will contain the user name and role associated with the policy match. The logging of user name and roles is enabled only for security policies that provide UAC enforcement. For security policies without UAC enforcement, the user name and user role fields are displayed as N/A. The user name is displayed as unauthenticated user and user role is displayed as N/A, if the device cannot retrieve information for that session because there is no authentication table entry for that session or because logging of this information is disabled. The user role field in the log contains the list of all the roles performed by the user if match criteria is specific, authenticated user, or any, and the user name field in the log contains the correct user name. The user role field in the log will contain N/A if the match criteria and the user name field in the log contain unauthenticated user or unknown user.

If you enable AppTrack for a zone and specify a `session-update-interval` time, whenever a packet is received, AppTrack checks whether the time since the start of the session or since the last update is greater than the update interval. If so, AppTrack updates the counts and sends an update message to the host. If a short-lived session starts and ends within the update interval, AppTrack generates a message only at session close.

When you want the initial update message to be sent earlier than the specified update interval, use the `first-update-interval`. The `first-update-interval` lets you enter a shorter interval for the first update only. Alternatively, you can generate the initial update message at session start by using the `first-update` option.

The close message updates the statistics for the last time and provides an explanation for the session closure. The following codes are used:

`TCP RST`	RST received from either end.
`TCP FIN`	FIN received from either end.
`Response received`	Response received for a packet request (such as `icmp req-reply`).
`ICMP error`	ICMP error received (such as `dest unreachable`).
`Aged out`	Session aged out.
`ALG`	ALG closed the session.
`IDP`	IDP closed the session.
`Parent closed`	Parent session closed.
`CLI`	Session cleared by a CLI statement.
`Policy delete`	Policy marked for deletion.

Benefits of Application Tracking

- Provides visibility into the types of applications traversing through your security device.
- Enables you to gain insight into permitted applications and the risk they might pose.
- Assists in managing bandwidth, reports active users and applications.

Application Tracking Log Messages Fields

Starting from Junos OS Release 15.1X49-D100, AppTrack session create, session close, and volume update logs include a new field called *destination interface*. You can use the `destination interface` field to see which egress interface is selected for the session when a advanced policy-based routing (APBR) is applied to that session and AppTrack is enabled and configured within any logical system.

Starting from Junos OS Release 15.1X49-D100, a new AppTrack log for route update is added to include APBR profile, rule, and routing instance details. When APBR is applied to a session, the new log is generated and the AppTrack session counter is updated to indicate the number of times a new route update log is generated. The AppTrack session close log is also updated to include APBR profile, rule, and routing instance details.

Starting from Junos OS Release 17.4R1, AppTrack session create, session close, and volume update logs include the new fields `category` and `subcategory`. These fields provide general information about the application attributes. For example, the `category` field specifies the technology of the application (web, infrastructure) and `subcategory` field specifies the subcategory of the application (for example, social networking, news, and advertisements).

Because category and subcategory are not applicable for a custom application, the AppTrack log messages present the category as `custom application` and the subcategory as `N/A`.

For unknown applications, both category and subcategories are logged as `N/A`.

Examples of the log messages in structured syslog format:

```
APPTRACK_SESSION_CREATE user@host.1.1.1.2.129 source-address="4.0.0.1" source-port="48873" destination-
address="5.0.0.1" destination-port="80" service-name="junos-http" application="UNKNOWN" nested-
application="UNKNOWN" nat-source-address="4.0.0.1" nat-source-port="48873" nat-destination-address="5.0.0.1" nat-
destination-port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="permit-all"
source-zone-name="trust" destination-zone-name="untrust" session-id-32="32" username="user1" roles="DEPT1"
encrypted="UNKNOWN" destination-interface-name=”ge-0/0/0” category=”N/A” sub-category=”N/A”]
```

```
APPTRACK_SESSION_CLOSE [junos@2636.1.1.1.2.129 reason="TCP CLIENT RST" source-address="4.0.0.1" source-
port="48873" destination-address="5.0.0.1" destination-port="80" service-name="junos-http" application="HTTP"
nested-application="UNKNOWN" nat-source-address="4.0.0.1" nat-source-port="48873" nat-destination-
address="5.0.0.1" nat-destination-port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6"
policy-name="permit-all" source-zone-name="trust" destination-zone-name="untrust" session-id-32="32" packets-from-
client="5" bytes-from-client="392" packets-from-server="3" bytes-from-server="646" elapsed-time="3"
username="user1" roles="DEPT1" encrypted="No" routing-instance=“default” destination-interface-name=”st0.0”
category=” Web” sub-category=”N/A”]
```

```
APPTRACK_SESSION_VOL_UPDATE [user@host.1.1.1.2.129 source-address="4.0.0.1" source-port="33040" destination-
address="5.0.0.1" destination-port="80" service-name="junos-http" application="HTTP" nested-application="FACEBOOK-
SOCIALRSS" nat-source-address="4.0.0.1" nat-source-port="33040" nat-destination-address="5.0.0.1" nat-destination-
port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="permit-all" source-zone-
name="trust" destination-zone-name="untrust" session-id-32="28" packets-from-client="371" bytes-from-
client="19592" packets-from-server="584" bytes-from-server="686432" elapsed-time="60" username="user1"
roles="DEPT1" encrypted="No" destination-interface-name=”st0.0” category=” Web” sub-category=”Social-
Networking”]
```

```
APPTRACK_SESSION_ROUTE_UPDATE [user@host.1.1.1.2.129 source-address="4.0.0.1" source-port="33040" destination-
address="5.0.0.1" destination-port="80" service-name="junos-http" application="HTTP" nested-application="FACEBOOK-
SOCIALRSS" nat-source-address="4.0.0.1" nat-source-port="33040" nat-destination-address="5.0.0.1" nat-destination-
port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="permit-all" source-zone-
name="trust" destination-zone-name="untrust" session-id-32="28" username="user1" roles="DEPT1" encrypted="No"
profile-name=”pf1” rule-name=”facebook1” routing-instance=”instance1” destination-interface-name=”st0.0”
category=”Web” sub-category=”Social-Networking”]
```

Starting in Junos OS Release 18.4R1 and Junos OS Release 18.3R2, in the APPTRACK_SESSION_ROUTE_UPDATE log, the `encrypted` field displays the value as `N/A` as shown in the following sample:

```
APPTRACK_SESSION_ROUTE_UPDATE [junos@2636.1.1.1.2.129 source-address="4.0.0.1" source-port="251" destination-
address="5.0.0.1" destination-port="250" service-name="None" application="HTTP" nested-application="UNKNOWN" nat-
source-address="4.0.0.1" nat-source-port="251" nat-destination-address="5.0.0.1" nat-destination-port="250" src-
nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="1" source-zone-name="trust" destination-
zone-name="untrust" session-id-32="866" username="N/A" roles="N/A" encrypted="N/A" profile-name="profile1"
rule-name="rule1" routing-instance="RI1" destination-interface-name="ge-0/0/2.0" category="Web" subcategory="N/A"
apbr-policy-name="sla1" webfilter-category="N/A"]
```

Starting in Junos OS Release 18.4R1, in the APPTRACK_SESSION_CLOSE and APPTRACK_SESSION_CLOSE_LS log includes the multipath rule name as shown in the following sample:

```
2018-10-25T01:00:18.179-07:00 multihome-spoke RT_FLOW - APPTRACK_SESSION_CLOSE [junos@2636.1.1.1.2.129
reason="idle Timeout" source-address="19.0.0.2" source-port="34880" destination-address="9.0.0.2" destination-
port="80" service-name="junos-http" application="HTTP" nested-application="GOOGLE-GEN" nat-source-
address="19.0.0.2" nat-source-port="34880" nat-destination-address="9.0.0.2" nat-destination-port="80" src-nat-
rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="1" source-zone-name="trust" destination-zone-
name="untrust1" session-id-32="9625" packets-from-client="347" bytes-from-client="18199" packets-from-server="388"
bytes-from-server="131928" elapsed-time="411" username="N/A" roles="N/A" encrypted="No" profile-name="apbr1" rule-
name="rule1" routing-instance="TC1_VPN" destination-interface-name="gr-0/0/0.4" uplink-incoming-interface-name=""
uplink-tx-bytes="0" uplink-rx-bytes="0" multipath-rule-name="multi1"]
```

Starting from Junos OS Release 18.2R1, AppTrack session close logs include new fields to record the packet bytes transmitted and received through the uplink interfaces. The packet bytes transmitted and received through the uplink interfaces are reported by `uplink-tx-bytes`, `uplink-rx-bytes`, and `uplink-incoming-interface-name` fields.

Example:

```
APPTRACK_SESSION_CLOSE [user@host.1.1.1.2.137 reason="TCP FIN" source-address="4.0.0.1" source-port="40297"
destination-address="5.0.0.1" destination-port="110" service-name="junos-pop3" application="POP3" nested-
application="UNKNOWN" nat-source-address="4.0.0.1" nat-source-port="40297" nat-destination-address="5.0.0.1" nat-
destination-port="110" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="permit-all"
source-zone-name="UNTRUST" destination-zone-name="TRUST" session-id-32="81" packets-from-client="7" bytes-from-
client="1959" packets-from-server="6" bytes-from-server="68643" elapsed-time="130" username="N/A" roles="N/A"
encrypted="No" profile-name="pf1" rule-name="facebook1" routing-instance="instance1" destination-interface-
name="gr-0/0/0.0" uplink-tx-bytes="1959" uplink-rx-bytes="68643" uplink-incoming-interface-
name="gr-0/0/0.0"]
```

Starting from Junos OS Release 18.2R1, the following new messages are added. These messages provide information such as active and passive metric report, switching of application traffic path as shown in the following samples:

```
APPQOE_BEST_PATH_SELECTED [junos@2636.1.1.1.2.129 source-address="20.1.1.1" source-port="47335" destination-
address="151.101.9.67" destination-port="443" apbr-profile="apbrProf1" apbr-rule="rule1" application="HTTP"
nested-application="CNN" group-name="N/A" service-name="junos-https" protocol-id="6" source-zone-name="trust"
destination-zone-name="untrust" session-id-32="611" username="N/A" roles="N/A" routing-instance="ri3" destination-
interface-name="gr-0/0/0.2" ip-dscp="0" sla-rule="SLA1" elapsed-time="2" bytes-from-client="675" bytes-from-
server="0" packets-from-client="7" packets-from-server="0" previous-interface="gr-0/0/0.2" active-probe-
params="PP1" destination-group-name="p1"]
```

```
APPQOE_PASSIVE_SLA_METRIC_REPORT [junos@2636.1.1.1.2.129 source-address="20.1.1.1" source-port="47335"
destination-address="151.101.9.67" destination-port="443" apbr-profile="apbrProf1" apbr-rule="rule1"
application="HTTP" nested-application="CNN" group-name="N/A" service-name="junos-https" protocol-id="6" source-
zone-name="trust" destination-zone-name="untrust" session-id-32="611" username="N/A" roles="N/A" routing-
instance="ri3" destination-interface-name="gr-0/0/0.2" ip-dscp="0" sla-rule="SLA1" ingress-jitter="0" egress-
jitter="0" rtt-jitter="0" rtt="0" pkt-loss="0" bytes-from-client="1073" bytes-from-server="6011" packets-from-
client="12" packets-from-server="13" monitoring-time="990" active-probe-params="PP1" destination-group-name="p1"]
```

```
APPQOE_SLA_METRIC_VIOLATION [junos@2636.1.1.1.2.129 source-address="20.1.1.1" source-port="35264" destination-
address="151.101.193.67" destination-port="443" apbr-profile="apbrProf1" apbr-rule="rule1" application="HTTP"
nested-application="CNN" group-name="N/A" service-name="junos-https" protocol-id="6" source-zone-name="trust"
destination-zone-name="untrust" session-id-32="614" username="N/A" roles="N/A" routing-instance="ri3" destination-
interface-name="gr-0/0/0.2" ip-dscp="0" sla-rule="SLA1" ingress-jitter="104" egress-jitter="7" rtt-jitter="97"
rtt="1142" pkt-loss="0" target-jitter-type="2" target-jitter="20000" target-rtt="500" target-pkt-loss="1"
violation-reason="1" jitter-violation-count="0" pkt-loss-violation-count="0" rtt-violation-count="1" violation-
duration="0" bytes-from-client="2476" bytes-from-server="163993" packets-from-client="48" packets-from-
server="150" monitoring-time="948" active-probe-params="PP1" destination-group-name="p1"]
```

```
APPQOE_ACTIVE_SLA_METRIC_REPORT [junos@2636.1.1.1.2.129 source-address="6.1.1.2" source-port="36051" destination-
address="6.1.1.1" destination-port="36050" application="UDP" protocol-id="17" destination-zone-name="untrust"
routing-instance="ri3" destination-interface-name="gr-0/0/0.3" ip-dscp="128" ingress-jitter="26" egress-
jitter="31" rtt-jitter="8" rtt="2383" pkt-loss="0" bytes-from-client="870240" bytes-from-server="425280" packets-
from-client="4440" packets-from-server="4430" monitoring-time="30" active-probe-params="PP1" destination-group-
name="p1"]
```

Starting in Junos OS Release 15.1X49-D170, AppTrack session create, session close, route update, and volume update logs are enhanced to include VRF name for both source VRF and destination-VRF.

```
RT_FLOW - APPTRACK_SESSION_ROUTE_UPDATE [junos@2636.1.1.1.2.129 source-address="1.3.0.10" source-port="990"
destination-address="8.3.0.10" destination-port="8080" service-name="None" application="HTTP" nested-
application="UNKNOWN" nat-source-address="1.3.0.10" nat-source-port="990" nat-destination-address="8.3.0.10" nat-
destination-port="8080" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="1" source-
zone-name="trust_lan2" destination-zone-name="sdwan" session-id-32="432399" username="N/A" roles="N/A"
encrypted="No" profile-name="p2" rule-name="r1" routing-instance="Default_VPN_LAN2" destination-interface-
name="gr-0/0/0.0" source-l3vpn-vrf-group-name="vpn-A" destination-l3vpn-vrf-group-name="vpn-A"]
```

```
RT_FLOW - APPTRACK_SESSION_CREATE [junos@2636.1.1.1.2.129 source-address="1.3.0.10" source-port="990" destination-
address="8.3.0.10" destination-port="8080" service-name="None" application="HTTP" nested-application="UNKNOWN"
nat-source-address="1.3.0.10" nat-source-port="990" nat-destination-address="8.3.0.10" nat-destination-port="8080"
src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="1" source-zone-name="trust_lan2"
destination-zone-name="sdwan" session-id-32="432399" username="N/A" roles="N/A" encrypted="No" destination-
interface-name="gr-0/0/0.0" source-l3vpn-vrf-group-name="vpn-A" destination-l3vpn-vrf-group-name="vpn-A’"]
```

```
RT_FLOW - APPTRACK_SESSION_VOL_UPDATE [junos@2636.1.1.1.2.129 source-address="4.0.0.1" source-port="34219"
destination-address="5.0.0.1" destination-port="80" service-name="junos-http" application="HTTP" nested-
application="UNKNOWN" nat-source-address="4.0.0.1" nat-source-port="34219" nat-destination-address="5.0.0.1" nat-
destination-port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="policy1"
source-zone-name="trust" destination-zone-name="untrust" session-id-32="4" packets-from-client="6" bytes-from-
client="425" packets-from-server="5" bytes-from-server="561" elapsed-time="1" username="N/A" roles="N/A"
encrypted="No" profile-name="p1" rule-name="r1" routing-instance="default" destination-interface-name="ge-0/0/1.0"
source-l3vpn-vrf-group-name="vpn-A" destination-l3vpn-vrf-group-name="vpn-A"]
```

Starting in Junos OS Release 19.1R1, the session close logs include new field source identity to check the session create log and session close log with user name and roles. The new messages provide information such as user name and roles as shown in the following sample:

```
RT_FLOW - APPTRACK_SESSION_CLOSE [junos@2636.1.1.1.2.129 reason="TCP FIN" source-address="4.0.0.1" source-
port="34219" destination-address="5.0.0.1" destination-port="80" service-name="junos-http" application="HTTP"
nested-application="UNKNOWN" nat-source-address="4.0.0.1" nat-source-port="34219" nat-destination-
address="5.0.0.1" nat-destination-port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6"
policy-name="policy1" source-zone-name="trust" destination-zone-name="untrust" session-id-32="4" packets-from-
client="6" bytes-from-client="425" packets-from-server="5" bytes-from-server="561" elapsed-time="1" username="N/A"
roles="N/A" encrypted="No" profile-name="p1" rule-name="r1" routing-instance="default" destination-interface-
name="ge-0/0/1.0" uplink-incoming-interface-name="" uplink-tx-bytes="0" uplink-rx-bytes="0" multipath-rule-
name="N/A" source-l3vpn-vrf-group-name="vpn-A" destination-l3vpn-vrf-group-name="vpn-A"]
```

A new syslog message `RT_FLOW_NEXTHOP_CHANGE` is generated whenever there is a change in the route or in the next-hop on the APBR and AppTrack enabled sessions.

> **NOTE**: In Junos OS release prior to 20.2R3, 20.3R2, 20.4R2, and 21.1R1, when an application is not identified by APBR (APBR interest check) and later it is identified by JDPI for first packet of the session, a syslog (RT_FLOW_NEXTHOP_CHANGE log) is generated. You can ignore the log message.

```
RT_FLOW_NEXTHOP_CHANGE [junos@2636.1.1.1.2.129 source-address="4.1.0.1" source-port="43540" destination-address="5.1.0.1" destination-port="7000" service-name="None" application="JNPR-UDPSVR-ADDR" nested-application="UNKNOWN" nat-source-address="4.1.0.1" nat-source-port="43540" nat-destination-address="5.1.0.1" nat-destination-port="7000" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="17" policy-name="1" source-zone-name="trust" destination-zone-name="untrust" session-id-32="2" packets-from-client="1" bytes-from-client="105" packets-from-server="0" bytes-from-server="0" elapsed-time="0" username="N/A" roles="N/A" encrypted="No" profile-name="profile1" rule-name="rule1" routing-instance="RI1" destination-interface-name="ge-0/0/1.0" last-destination-interface-name="ge-0/0/4.0" uplink-incoming-interface-name="" last-incoming-interface-name="N/A" uplink-tx-bytes="0" uplink-rx-bytes="0" apbr-policy-name="N/A" dscp-value="N/A" apbr-rule-type="application"]
```

Starting in Junos OS Release 19.3R1, AppTrack session logs such as session close, volume update, route update, and RT_FLOW_NEXTHOP_CHANGE include `dscp-value` and `apbr-rule-type` options.

-
  ```
  APPTRACK_SESSION_CLOSE [junos@2636.1.1.1.2.129 reason="TCP CLIENT RST" source-address="4.0.0.1" source-port="48873" destination-address="5.0.0.1" destination-port="80" service-name="junos-http" application="UNKNOWN" nested-application="UNKNOWN" nat-source-address="4.0.0.1" nat-source-port="48873" nat-destination-address="5.0.0.1" nat-destination-port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="permit-all" source-zone-name="trust" destination-zone-name="untrust" session-id-32="32" packets-from-client="5" bytes-from-client="392" packets-from-server="3" bytes-from-server="646" elapsed-time="3" username="user1" roles="DEPT1" encrypted="No" destination-interface-name=”st0.0” dscp-value=”13” apbr-rule-type=”dscp”]
  ```

-
  ```
  APPTRACK_SESSION_ROUTE_UPDATE [junos@2636.1.1.1.2.129 source-address="4.0.0.1" source-port="33040" destination-address="5.0.0.1" destination-port="80" service-name="junos-http" application="HTTP" nested-application="FACEBOOK-SOCIALRSS" nat-source-address="4.0.0.1" nat-source-port="33040" nat-destination-address="5.0.0.1" nat-destination-port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="permit-all" source-zone-name="trust" destination-zone-name="untrust" session-id-32="28" username="user1" roles="DEPT1" encrypted="No" profile-name="pf1" rule-name=”facebook1” routing-instance=”instance1” destination-interface-name=”st0.0” dscp-value=”13” apbr-rule-type=”application-dscp”]
  ```

-
  ```
  APPTRACK_SESSION_VOL_UPDATE [junos@2636.1.1.1.2.129 source-address="4.0.0.1" source-port="33040" destination-address="5.0.0.1" destination-port="80" service-name="junos-http" application="HTTP" nested-application="FACEBOOK-SOCIALRSS" nat-source-address="4.0.0.1" nat-source-port="33040" nat-destination-address="5.0.0.1" nat-destination-port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="permit-all" source-zone-name="trust" destination-zone-name="untrust" session-id-32="28" packets-from-client="371" bytes-from-client="19592" packets-from-server="584" bytes-from-server="686432" elapsed-
  ```

time="60" username="user1" roles="DEPT1" encrypted="No" destination-interface-name=”st0.0” dscp-value=”13” apbr-rule-type=”application-dscp”]

- RT_FLOW_NEXTHOP_CHANGE [junos@2636.1.1.1.2.129 source-address="4.0.0.1" source-port="1999" destination-address="157.240.23.35" destination-port="80" service-name="junos-http" application="UNKNOWN" nested-application="UNKNOWN" nat-source-address="4.0.0.1" nat-source-port="1999" nat-destination-address="157.240.23.35" nat-destination-port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="1" source-zone-name="trust" destination-zone-name="untrust" session-id-32="3287" packets-from-client="1" bytes-from-client="60" packets-from-server="0" bytes-from-server="0" elapsed-time="0" username="N/A" roles="N/A" encrypted="No" profile-name="profile1" rule-name="rule1" routing-instance="RI1" destination-interface-name="ge-0/0/1.0" last-destination-interface-name="ge-0/0/4.0" uplink-incoming-interface-name="" last-incoming-interface-name="N/A" uplink-tx-bytes="0" uplink-rx-bytes="0" apbr-policy-name="sla1" dscp-value=”13” apbr-rule-type=”dscp”]

Starting in Junos OS Release 20.1R1, AppTrack session logs such as session close, volume update, route update include `apbr-rule-type` options.

- APPTRACK_SESSION_VOL_UPDATE [junos@2636.1.1.1.2.129 source-address="4.0.0.1" source-port="33040" destination-address="5.0.0.1" destination-port="80" service-name="junos-http" application="HTTP" nested-application="FACEBOOK-SOCIALRSS" nat-source-address="4.0.0.1" nat-source-port="33040" nat-destination-address="5.0.0.1" nat-destination-port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="permit-all" source-zone-name="trust" destination-zone-name="untrust" session-id-32="28" packets-from-client="371" bytes-from-client="19592" packets-from-server="584" bytes-from-server="686432" elapsed-time="60" username="user1" roles="DEPT1" encrypted="No" destination-interface-name=”st0.0” apbr-rule-type=”default”]

- APPTRACK_SESSION_ROUTE_UPDATE [junos@2636.1.1.1.2.129 source-address="4.0.0.1" source-port="33040" destination-address="5.0.0.1" destination-port="80" service-name="junos-http" application="HTTP" nested-application="FACEBOOK-SOCIALRSS" nat-source-address="4.0.0.1" nat-source-port="33040" nat-destination-address="5.0.0.1" nat-destination-port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="permit-all" source-zone-name="trust" destination-zone-name="untrust" session-id-32="28" username="user1" roles="DEPT1" encrypted="No" profile-name=”pf1” rule-name=”facebook1” routing-instance=”instance1” destination-interface-name=”st0.0” apbr-rule-type=”default”]

- APPTRACK_SESSION_CLOSE [junos@2636.1.1.1.2.129 reason="TCP CLIENT RST" source-address="4.0.0.1" source-port="48873" destination-address="5.0.0.1" destination-port="80" service-name="junos-http" application="UNKNOWN" nested-application="UNKNOWN" nat-source-address="4.0.0.1" nat-source-port="48873" nat-destination-address="5.0.0.1" nat-destination-port="80" src-nat-rule-name="N/A" dst-nat-rule-name="N/A" protocol-id="6" policy-name="permit-all" source-zone-name="trust" destination-zone-name="untrust" session-id-32="32" packets-from-client="5" bytes-from-client="392" packets-from-server="3" bytes-from-server="646" elapsed-time="3" username="user1" roles="DEPT1" encrypted="No" destination-interface-name=”st0.0” apbr-rule-type=”default”]

Starting in Junos OS Release 20.4R1, AppTrack session logs for AppQoE such as best path selected, SLA metric violation, SLA metric reports are updated.

-
  ```
  APPQOE_APP_BEST_PATH_SELECTED [junos@2636.1.1.1.2.129 apbr-profile="apbr1" apbr-rule="rule1" application="ANY" other-app="N/A" group-name="N/A" routing-instance="TC1_VPN" previous-interface="N/A" destination-interface-name="gr-0/0/0.0" sla-rule="sla1" active-probe-params="probe1" destination-group-name="site1" reason="app detected" session-count="1" violation-duration="0" ip-dscp="255" selection-criteria=“default” "server-ip=”10.1.1.1” url=”salesforce.com”]
  ```

-
  ```
  APPQOE_APP_SLA_METRIC_VIOLATION [junos@2636.1.1.1.2.129 apbr-profile=" apbr1" apbr-rule="rule1" application="ANY" other-app="N/A" group-name="N/A" routing-instance="ri3" destination-interface-name="gr-0/0/0.0" sla-rule="SLA1" ingress-jitter="4294967295" egress-jitter="4294967295" rtt-jitter="1355" rtt="5537" pkt-loss="0" target-jitter-type="2" target-jitter="20000" target-rtt="1000" target-pkt-loss="1" violation-reason="1" violation-duration="20" active-probe-params="PP1" destination-group-name="p1" "server-ip=”10.1.1.1” url=”salesforce.com”]
  ```

-
  ```
  APPQOE_ACTIVE_SLA_METRIC_REPORT [junos@2636.1.1.1.2.129 source-address="40.1.1.2" source-port="10001" destination-address="40.1.1.1" destination-port="80" destination-zone-name="untrust1" routing-instance="transit" destination-interface-name="" ip-dscp="6" ingress-jitter="4294967295" egress-jitter="4294967295" rtt-jitter="1345" rtt="4294967295" pkt-loss="100" monitoring-time="29126" active-probe-params="probe1" destination-group-name="site1" forwarding-class="network-control" loss-priority="low" active-probe-type="http head"]
  ```

Starting in Junos OS Release 21.2R1, for an application profile without SLA metric, the AppQoE generates only the `APPQOE_APP_BEST_PATH_SELECTED` log. In the `APPQOE_APP_BEST_PATH_SELECTED` log, the `active-probe-params` field displays `N/A` and the violation duration field displays `N/A`. The `APPQOE_APP_BEST_PATH_SELECTED` log has the new fields such as `previous-link-tag`, `previous-link-priority`, `destination-link-tag`, and `destination-link-priority` as shown in the following samples. When the `reason` is `app detected`, then the `previous-link-tag` field displays `N/A` and the `previous-link-priority` field displays `0`.

- Application independent profile without SLA metric considerations.

  ```
  APPQOE_APP_BEST_PATH_SELECTED [junos@2636.1.1.1.2.129 apbr-profile="apbr1" apbr-rule="rule1" application="ANY" other-app="N/A" group-name="N/A" routing-instance="TC1_VPN" previous-interface="gr-0/0/0.1" destination-interface-name="gr-0/0/0.3" sla-rule="sla1" active-probe-params="N/A" destination-group-name="site1" reason="switch to high priority link" session-count="2" violation-duration="N/A" ip-dscp="255" selection-criteria="default" forwarding-nexthop-id="262142" server-ip="0.0.0.0" server-url="N/A" previous-link-tag=”ISP1” previous-link-priority=”100” destination-link-tag=”ISP1” destination-link-priority=”50”]
  ```

  For application independent profile the application field displays as `ANY`.

- Application based profile without SLA metrics considerations.

  ```
  APPQOE_APP_BEST_PATH_SELECTED [junos@2636.1.1.1.2.129 apbr-profile="apbr1" apbr-rule="rule1" application="SSH" other-app="N/A" group-name="N/A" routing-instance="TC1_VPN" previous-interface="gr-0/0/0.1" destination-interface-name="gr-0/0/0.3" sla-rule="sla1" active-probe-params="N/A" destination-group-name="site1" reason="switch to high priority link" session-count="2" violation-duration="N/A" ip-dscp="255" selection-criteria="application" forwarding-nexthop-id="262142" server-ip="0.0.0.0" server-url="N/A" previous-link-tag=”ISP1” previous-link-priority=”100” destination-link-tag=”ISP1” destination-link-priority=”50”]
  ```

- Application independent profile with SLA metric considerations and without violation reported.

```
APPQOE_APP_BEST_PATH_SELECTED [junos@2636.1.1.1.2.129 apbr-profile="apbr1" apbr-rule="rule1" application="ANY" other-app="N/A" group-name="N/A" routing-instance="TC1_VPN" previous-interface="gr-0/0/0.1" destination-interface-name="gr-0/0/0.3" sla-rule="sla1" active-probe-params="probe1" destination-group-name="site1" reason="switch to high priority link" session-count="2" violation-duration=”180” ip-dscp="255" selection-criteria="default" forwarding-nexthop-id="262142" server-ip="0.0.0.0" server-url="N/A" previous-link-priority=”100” destination-link-tag=”ISP2” destination-link-priority=”50”]
```

- Application independent profile with SLA metric considerations and with violation reported.

```
APPQOE_APP_BEST_PATH_SELECTED [junos@2636.1.1.1.2.129 apbr-profile="apbr1" apbr-rule="rule1" application="ANY" other-app="N/A" group-name="N/A" routing-instance="TC1_VPN" previous-interface="gr-0/0/0.1" destination-interface-name="gr-0/0/0.3" sla-rule="sla1" active-probe-params="probe1" destination-group-name="site1" reason="sla violated" session-count="2" violation-duration=”180” ip-dscp="255" selection-criteria="default" forwarding-nexthop-id="262142" server-ip="0.0.0.0" server-url="N/A" previous-link-priority=”100” destination-link-tag=”ISP2” destination-link-priority=”50”]
```

```
APPQOE_APP_SLA_METRIC_VIOLATION [junos@2636.1.1.1.2.129 apbr-profile="apbr1" apbr-rule="rule1" application="ANY" other-app="N/A" group-name="N/A" routing-instance="TC1_VPN" destination-interface-name="gr-0/0/0.1" sla-rule="sla1" ingress-jitter="253" egress-jitter="252340" rtt-jitter="252593" rtt="251321" pkt-loss="0" target-jitter-type="2" target-jitter="25000" target-rtt="200000" target-pkt-loss="15" violation-reason="3" active-probe-params="probe1" destination-group-name="site1" ip-dscp="255" selection-criteria="default"]
```

```
APPQOE_APP_PASSIVE_SLA_METRIC_REPORT [junos@2636.1.1.1.2.129 apbr-profile="apbr1" apbr-rule="rule1" application="ANY" other-app="N/A" group-name="N/A" routing-instance="TC1_VPN" destination-interface-name="gr-0/0/0.1" sla-rule="sla1" ingress-jitter="109" egress-jitter="72" rtt-jitter="102" rtt="63674" pkt-loss="0" min-ingress-jitter="1" min-egress-jitter="1" min-rtt-jitter="1" min-rtt="793" min-pkt-loss="0" max-ingress-jitter="448" max-egress-jitter="252340" max-rtt-jitter="252593" max-rtt="253784" max-pkt-loss="0" probe-count="122" monitoring-time="59882" active-probe-params="probe1" destination-group-name="site1" ip-dscp="255" selection-criteria="default"]
```

- Application based profile with SLA metric considerations and without violation reported.

```
APPQOE_APP_BEST_PATH_SELECTED [junos@2636.1.1.1.2.129 apbr-profile="apbr1" apbr-rule="rule1" application="SSH" other-app="N/A" group-name="N/A" routing-instance="TC1_VPN" previous-interface="gr-0/0/0.2" destination-interface-name="gr-0/0/0.3" sla-rule="sla1" active-probe-params="probe1" destination-group-name="site1" reason="switch to high priority link" session-count="2" violation-duration="180" ip-dscp="255" selection-criteria="application" forwarding-nexthop-id="262142" server-ip="0.0.0.0" server-url="N/A" previous-link-tag=”ISP2” previous-link-priority=”70” destination-link-tag=”ISP2” destination-link-priority=”30”]
```

Application based profile with SLA metric considerations and with violation reported.

```
APPQOE_APP_BEST_PATH_SELECTED [junos@2636.1.1.1.2.129 apbr-profile="apbr1" apbr-rule="rule1" application="SSH" other-app="N/A" group-name="N/A" routing-instance="TC1_VPN" previous-interface="gr-0/0/0.2" destination-
interface-name="gr-0/0/0.3" sla-rule="sla1" active-probe-params="probe1" destination-group-name="site1"
reason="sla violated" session-count="2" violation-duration="180" ip-dscp="255" selection-
criteria="application" forwarding-nexthop-id="262142" server-ip="0.0.0.0" server-url="N/A" previous-link-
tag=”ISP2” previous-link-priority=”70” destination-link-tag=”ISP2” destination-link-priority=”30”]
```

```
APPQOE_APP_PASSIVE_SLA_METRIC_REPORT [junos@2636.1.1.1.2.129 apbr-profile="apbr1" apbr-rule="rule1"
application="SSH" other-app="N/A" group-name="N/A" routing-instance="TC1_VPN" destination-interface-
name="gr-0/0/0.1" sla-rule="sla1" ingress-jitter="109" egress-jitter="72" rtt-jitter="102" rtt="63674" pkt-
loss="0" min-ingress-jitter="1" min-egress-jitter="1" min-rtt-jitter="1" min-rtt="793" min-pkt-loss="0" max-
ingress-jitter="448" max-egress-jitter="252340" max-rtt-jitter="252593" max-rtt="253784" max-pkt-loss="0"
probe-count="122" monitoring-time="59882" active-probe-params="probe1" destination-group-name="site1" ip-
dscp="255" selection-criteria="application"]
```

```
APPQOE_APP_SLA_METRIC_VIOLATION [junos@2636.1.1.1.2.129 apbr-profile="apbr1" apbr-rule="rule1"
application="SSH" other-app="N/A" group-name="N/A" routing-instance="TC1_VPN" destination-interface-
name="gr-0/0/0.1" sla-rule="sla1" ingress-jitter="253" egress-jitter="252340" rtt-jitter="252593" rtt="251321"
pkt-loss="0" target-jitter-type="2" target-jitter="25000" target-rtt="200000" target-pkt-loss="15" violation-
reason="3" active-probe-params="probe1" destination-group-name="site1" ip-dscp="255" selection-
criteria="application"]
```

Consider a scenario where an SRX Series Firewall is operating in chassis cluster mode, and the AppQoE configuration includes SaaS probes and violation count value configured as 1. When application traffic switches the route path across the node, violation syslog message is generated on both primary and backup nodes. You can ignore the syslog generated on the node that is hosting the current path.

- When the link affinity is configured as “loose” and if the application traffic switches from a preferred-link to a non-preferred-link, and that non-preferred-link has the higher priority, the system log message logs the reason as "switch to higher priority".

  Example:

  ```
  RT_FLOW - APPQOE_APP_BEST_PATH_SELECTED [apbr-profile="apbr1" apbr-rule="rule1" application="YAHOO" other-
  app="N/A" group-name="N/A" routing-instance="TC1_VPN" previous-interface="gr-0/0/0.0" destination-interface-
  name="gr-0/0/0.2" sla-rule="sla1" active-probe-params="probe1" destination-group-name="site1" reason="switch
  to higher priority link" session-count="1" violation-duration="0" ip-dscp="255" selection-
  criteria="application" forwarding-nexthop-id="262149" server-ip="0.0.0.0" server-url="N/A" previous-link-
  tag="ISP1" previous-link-priority="10" destination-link-tag="ISP3" destination-link-priority="3"]
  ```

SEE ALSO

Example: Configuring Application Tracking | 178

Disabling Application Tracking | 188

Understanding Application Identification Techniques | 5

Example: Configuring Application Tracking

IN THIS SECTION

This example shows how to configure the AppTrack tracking tool so you can analyze the bandwidth usage of your network.

Requirements

Before you configure AppTrack, ensure that you have downloaded the application signature package, installed it, and verified that the application identification configuration is working properly. See "Downloading and Installing the Junos OS Application Signature Package Manually" on page 39 or "Downloading and Installing the Junos OS Application Signature Package As Part of the IDP Security Package" on page 44. Use the *show services application-identification status* command to verify the status.

Overview

Application identification is enabled by default and is automatically turned on when you configure the AppTrack, AppFW, or IDP service. The Juniper Secure Analytics (JSA) retrieves the data and provides flow-based application visibility. STRM includes the support for AppTrack Reporting and includes several predefined search templates and reports.

Configuration

IN THIS SECTION

This example shows how to enable application tracking for the security zone named trust. The first log message is to be generated when the session starts, and update messages should be sent every 4 minutes after that. A final message should be sent at session end.

The example also shows how to add the remote syslog device configuration to receive AppTrack log messages in sd-syslog format. The source IP address that is used when exporting security logs is 192.0.2.1, and the security logs are sent to the host located at address 192.0.2.2.

NOTE: J-Web pages for AppSecure Services are preliminary. We recommend using CLI for configuration of AppSecure features.

Procedure

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.

NOTE: Changing the `session-update-interval` and the `first-update-interval` is not necessary in most situations. The commands are included in this example to demonstrate their use.

```
user@host# set security log mode stream
user@host# set security log format sd-syslog
user@host# set security log source-address 192.0.2.1
user@host# set security log stream app-track-logs host 192.0.2.2
user@host#set security zones security-zone trust application-tracking
user@host#set security application-tracking session-update-interval 4
user@host#set security application-tracking first-update
```

NOTE: On SRX5600, and SRX5800 devices, if the syslog configuration does not specify a destination port, the default destination port will be the syslog port. If you specify a destination port in the syslog configuration, then that port will be used instead.

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 *CLI User Guide*.

To configure AppTrack:

1. Add the remote syslog device configuration to receive Apptrack messages in sd-syslog format.

```
[edit]
user@host# set security log mode stream
user@host# set security log format sd-syslog
user@host# set security log source-address 192.0.2.1
user@host# set security log stream app-track-logs host 192.0.2.2
```

2. Enable AppTrack for the security zone trust.

```
[edit]
user@host# set security zones security-zone trust application-tracking
```

3. (Optional) For this example, generate update messages every 4 minutes.

```
[edit]
user@host# set security application-tracking session-update-interval 4
```

The default interval between messages is 5 minutes. If a session starts and ends within this update interval, AppTrack generates one message at session close. However, if the session is long-lived, an update message is sent every 5 minutes. The `session-update-interval` *minutes* is configurable as shown in this step.

4. (Optional) For this example, generate the first message when the session starts.

```
[edit]
user@host# set security application-tracking first-update
```

By default, the first message is generated after the first session update interval elapses. To generate the first message at a different time than this, use the `first-update` option (generate the first message at session start) or the `first-update-interval` *minutes* option (generate the first message after the specified minutes). For example, enter the following command to generate the first message one minute after session start.

```
[edit]
user@host# set security application-tracking first-update-interval 1
```

NOTE: The `first-update` option and the `first-update-interval` *`minutes`* option are mutually exclusive. If you specify both, the `first-update-interval` value is ignored.

Once the first message has been generated, an update message is generated each time the session update interval is reached.

Results

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

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

```
[edit]
user@host# show security
```

```
...
application-tracking {
    first-update;
    session-update-interval 4;
}
log {
    mode stream;
    format sd-syslog;
    source-address192.0.2.2;
    stream app-track-logs {
        host {
            192.0.2.1;
        }
    }
}
...
```

```
[edit]
user@host# show security zones
...
security-zone trust {
    ...
    application-tracking;
}
```

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

Verification

IN THIS SECTION

Use the JSA product on the remote logging device to view the AppTrack log messages.

To confirm that the configuration is working properly, you can also perform these tasks on the device.

Reviewing AppTrack Statistics

Purpose

Review AppTrack statistics to view characteristics of the traffic being tracked.

Action

From operational mode, enter the `show services application-identification statistics applications` command.

```
user@host> show services application-identification statistics applications
```

```
Last Reset: 2012-02-14 21:23:45 UTC

Application    Sessions             Bytes    Encrypted
HTTP                  1              2291          Yes
HTTP                  1               942           No
SSL                   1              2291          Yes
unknown               1               100           No
unknown               1               100          Yes
```

NOTE: For more information on the `show services application-identification statistics applications` command, see *show services application-identification statistics applications*.

Verifying AppTrack Counter Values

Purpose

View the AppTrack counters periodically to monitor logging activity.

Action

From operational mode, enter the `show security application-tracking counters` command.

```
user@host> show security application-tracking counters
```

```
AVT counters:                               Value
 Session create messages             1
 Session close messages              1
 Session volume updates              0
 Failed messages                     0
```

Verifying Security Flow Session Statistics

Purpose

Compare byte and packet counts in logged messages with the session statistics from the `show security flow session` command output.

Action

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

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

```
Flow Sessions on FPC6 PIC0:

Session ID: 120000044, Policy name: policy-in-out/4, Timeout: 1796, Valid
In: 192.0.2.1/24 --> 198.51.100.0/21;tcp, If: ge-0/0/0.0, Pkts: 22, Bytes: 1032
Out: 198.51.100.0/24 --> 192.0.2.1//39075;tcp, If: ge-0/0/1.0, Pkts: 24, Bytes: 1442

Valid sessions: 1
Pending sessions: 0
Invalidated sessions: 0
Sessions in other states: 0
Total sessions: 1
```

Byte and packet totals in the session statistics should approximate the counts logged by AppTrack but might not be exactly the same. AppTrack counts only incoming bytes and packets. System-generated packets are not included in the total, and dropped packets are not deducted.

Verifying Application System Cache Statistics

Purpose

Compare cache statistics such as IP address, port, protocol, and service for an application from the `show services application-identification application-system-cache` command output.

Action

From operational mode, enter the `show services application-identification application-system-cache` command.

Verifying the Status of Application Identification Counter Values

Purpose

Compare session statistics for application identification counter values from the `show services application-identification counter` command output.

Action

From operational mode, enter the `show services application-identification counter` command.

SEE ALSO

Configuring Off-Box Binary Security Log Files

Understanding On-Box Logging and Reporting

log (Security Policies)

Example: Configuring Application Tracking When SSL Proxy Is Enabled

IN THIS SECTION

This example describes how AppTrack supports AppID functionality when SSL proxy is enabled.

Requirements

Before you begin:

- Create zones. See *Example: Creating Security Zones*.
- Create an SSL proxy profile that enables SSL proxy by means of a policy. See "Configuring SSL Forward Proxy" on page 387.

Overview

You can configure AppTrack either in the to or from zones. This example shows how to configure AppTrack in a to zone in a policy rule when SSL proxy is enabled.

Configuration

IN THIS SECTION

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 zones security-zone Z_1 application-tracking
set security policies from-zone Z_1 to-zone Z_2 policy policy1 match source-address any
set security policies from-zone Z_1 to-zone Z_2 policy policy1 match destination-address any
set security policies from-zone Z_1 to-zone Z_2 policy policy1 then permit application-services
ssl-proxy profile-name ssl-profile-1
set security policies from-zone Z_1 to-zone Z_2 policy policy1 then permit
```

Procedure

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 this example, you configure application tracking and permit application services in an SSL proxy profile configuration.

1. Configure application tracking in a to-zone (you can also configure using a from-zone).

```
[edit security policies]
user@host# set security zones security-zone Z_1 application-tracking
```

2. Configure SSL proxy profile.

```
[edit security policies from-zone Z_1 to-zone Z_2 policy policy1]
set match source-address any
set match destination-address any
set match application junos-https
set then permit application-services ssl-proxy profile-name ssl-profile-1
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 Z_1 to-zone Z_2 {
    policy policy1 {
        match {
            source-address any;
            destination-address any;
        }
        then {
            permit {
                application-services {
                    ssl-proxy {
                        profile-name ssl-profile-1;
                    }
                }
            }
        }
    }
}
```

NOTE: Verify that the configuration is working properly. Verification in AppTrack works similarly to verification in AppFW. See the verification section of "Example: Configuring Application Firewall When SSL Proxy Is Enabled" on page 161.

SEE ALSO

SSL Proxy Overview | 350

Disabling Application Tracking

Application tracking is enabled by default. You can disable application tracking without deleting the zone configuration.

To disable application tracking:

```
user@host# set security application-tracking disable
```

If application tracking has been previously disabled and you want to reenable it, delete the configuration statement that specifies disabling of application tracking:

```
user@host# delete security application-tracking disable
```

If you are finished configuring the device, commit the configuration.

To verify the configuration, enter the `show security application-tracking` command.

Release History Table

Release	Description
20.4R1	Starting in Junos OS Release 20.4R1, AppTrack session logs for AppQoE such as best path selected, SLA metric violation, SLA metric reports are updated.
19.3R1	Starting in Junos OS Release 19.3R1, AppTrack session logs such as session close, volume update, route update, and RT_FLOW_NEXTHOP_CHANGE include `dscp-value` and `apbr-rule-type` options.

19.3R1	Starting in Junos OS Release 20.1R1, AppTrack session logs such as session close, volume update, route update include `apbr-rule-type` options.
19.1R1	Starting in Junos OS Release 19.1R1, AppTrack session close logs include new field source identity to check the session create log and session close log with user name and roles.
18.2R1	Starting from Junos OS Release 18.2R1, AppTrack session close logs include new fields to record the packet bytes transmitted and received through the uplink interfaces.
18.2R1	Starting from Junos OS Release 18.2R1, new application tracking messages are added for AppQoE (application quality of experience).
17.4R1	Starting from Junos OS Release 17.4R1, AppTrack session create, session close, and volume update logs include the new fields `category` and `subcategory`
15.1X49-D170	Starting in Junos OS Release 15.1X49-D170, AppTrack session create, session close, route update, and volume update logs are enhanced to include VRF-name for both Source-VRF and Destination-VRF.
15.1X49-D100	Starting from Junos OS Release 15.1X49-D100, AppTrack session create, session close, and volume update logs include a new field called *destination interface*.
15.1X49-D100	Starting from Junos OS Release 15.1X49-D100, a new AppTrack log for route update is added to include APBR profile, rule, and routing instance details.

RELATED DOCUMENTATION

Application Identification | **5**

Application Firewall | **130**

Application QoS | **190**

Advanced Policy-Based Routing | **220**

SSL Proxy | **350**

Application QoS

IN THIS SECTION

AppQoS enable you to identify and control access to specific applications and provides the granularity of the stateful firewall rule base to match and enforce quality of service (QoS) at the application layer. For more information, see the following topics:

Understanding Application Quality of Service (AppQoS)

IN THIS SECTION

The application *quality of service* (AppQoS) feature expands the capability of Junos OS *class of service* (CoS) to include marking DSCP values based on Layer-7 application types, honoring application-based traffic through loss priority settings, and controlling transfer rates on egress PICs based on Layer-7 application types.

There are four ways to mark DSCP values on the security device:

- IDP attack action-based DSCP rewriters
- Layer 7 application-based DSCP rewriters
- ALG-based DSCP rewriters
- *Firewall filter*-based DSCP rewriters

IDP remarking is conducted at the ingress port based on IDP rules. Application remarking is conducted at the egress port based on application rules. Interface-based remarking also occurs at the egress port based on firewall filter rules. (See the Class of Service User Guide (Security Devices) for a detailed description of Junos OS CoS features.)

The remarking decisions of these three rewriters can be different. If a packet triggers all three, the method that takes precedence is based on how deep into the packet content the match is conducted. IDP remarking has precedence over application remarking which has precedence over interface-based remarking.

If a packet triggers both AppQoS and ALG-based DSCP rewriters, then AppQoS takes precedence over ALG-based DSCP rewriters.

The AppQoS DSCP rewriter conveys a packet's quality of service through both the forwarding class and a loss priority. The AppQoS rate-limiting parameters control the transmission speed and volume for its associated queues.

Benefit of Application QoS

AppQoS provides the ability to prioritize and meter the application traffic to provide better service to business-critical or high-priority application traffic.

Unique Forwarding Classes and Queue Assignments

The forwarding class provides three functions:

- Groups packets with like characteristics
- Assigns output queues
- Resolves conflicts with existing Junos OS firewall filter-based rewriters

Unique forwarding class names protect AppQoS remarking from being overwritten by interface-based *rewrite rules*. A firewall filter-based rewriter remarks a packet's DSCP value if the packet's forwarding class matches a class defined specifically for this rewriter. If the packet's forwarding class does not match any of the firewall filter-based rewriter's classes, the DSCP value is not remarked. To protect

AppQoS values from being overwritten, therefore, use forwarding class names that are unknown to the firewall filter-based rewriter.

Each forwarding class is assigned to an egress queue that provides the appropriate degree of enhanced or standard processing. Many forwarding classes can be assigned to a single queue. Therefore, any queues defined for the device can be used by IDP, AppQoS, and firewall filter-based rewriters. It is the forwarding class name, not the queue, that distinguishes the transmission priority. (See the Class of Service User Guide (Security Devices) for information about configuring queues and schedulers.)

For SRX5400, SRX5600, and SRX5800 devices, the AppQoS forwarding class names and queue assignments are defined with the `class-of-service` CLI configuration command:

```
[edit class-of-service]
user@host# set forwarding-classes class forwarding-class-name queue-num queue-number
```

For SRX300, SRX320, SRX340, SRX345, SRX380, SRX550M, SRX1500, SRX4100, SRX4200, and SRX4600 devices and vSRX Virtual Firewall instances, the AppQoS forwarding class names and queue assignments are defined with the `class-of-service` CLI configuration command:

```
[edit class-of-service]
user@host# set forwarding-classes queue queue-number forwarding-class-name
```

Application-Aware DSCP Code-Point and Loss Priority Settings

For AppQoS, traffic is grouped based on rules that associate a defined forwarding class with selected applications. The match criteria for the rule includes one or more applications. When traffic from a matching application encounters the rule, the rule action sets the forwarding class, and remarks the DSCP value and loss priority to values appropriate for the application.

A Differentiated Services (DiffServ) code point (DSCP) value is specified in the rule either by a 6-bit bitmap value or by a user-defined or default alias. Table 8 on page 192 provides a list of Junos OS default DSCP alias names and bitmap values.

Table 8: Standard CoS Aliases and Bit Values

Alias	Bit Value
ef	101110
af11	001010

Table 8: Standard CoS Aliases and Bit Values *(Continued)*

Alias	Bit Value
af12	001100
af13	001110
af21	010010
af22	010100
af23	010110
af31	011010
af32	011100
af33	011110
af41	100010
af42	100100
af43	100110
be	000000
cs1	001000
cs2	010000
cs3	011000

Table 8: Standard CoS Aliases and Bit Values *(Continued)*

Alias	Bit Value
cs4	100000
cs5	101000
nc1/cs6	110000
nc2/cs7	111000

See Default CoS Values and Aliases for more details.

The queue's scheduler uses the loss priority to control packet discard during periods of congestion by associating drop profiles with particular loss priority values. (See the Class of Service User Guide (Security Devices) for information about configuring queues and schedulers.)

The rule applies a loss priority to the traffic groups. A high loss priority means a high probability that the packet could be dropped during a period of congestion. Four levels of loss priority are available:

- `high`
- `medium-high`
- `medium-low`
- `low`

The rule set is defined in the `class-of-service application-traffic-control` configuration command:

```
[edit class-of-service]
user@host# set application-traffic-control rule-sets ruleset-name rule rule-name1  match
application application-name application-name ...
user@host# set application-traffic-control rule-sets ruleset-name rule rule-name1  match
application-group application-group-name application-group-name ...
user@host# set application-traffic-control rule-sets ruleset-name rule rule-name1  then
forwarding-class fc-name
user@host# set application-traffic-control rule-sets ruleset-name rule rule-name1  then  dscp-
code-point bitmap
user@host# set application-traffic-control rule-sets ruleset-name rule rule-name1  then  loss-
```

```
priority loss-pri-value
```

Rate Limiters and Profiles

When congestion occurs, AppQoS implements rate limiting on all egress PICs on the device. If packets exceed the assigned limitations, they are dropped. *Rate limiters* maintain a consistent level of throughput and packet loss sensitivity for different classes of traffic. All egress PICs employ the same rate-limiting scheme.

The total bandwidth of a PIC is about 10 Gbps. Rate-limiter hardware for the PIC can provision up to 2 Gbps. Therefore, the upper bandwidth limit for rate limiting is 2^{31} bps.

A rate-limiter profile defines the limitations. It is a unique combination of `bandwidth-limit` and `burst-size-limit` specifications. The `bandwidth-limit` defines the maximum number of kilobits per second that can traverse the port. The `burst-size-limit` defines the maximum number of bytes that can traverse the port in a single burst. The `burst-size-limit` reduces starvation of lower priority traffic by ensuring a finite size for each burst.

AppQoS allows up to 16 profiles and up to 1000 rate limiters per device. Multiple rate limiters can use the same profile. In the following example, five rate limiters are defined using two profiles:

Rate Limiter Name	Profile	
	bandwidth-limit	burst-size-limit
limiter-1	200	26000
limiter-2	200	26000
limiter-3	200	26000
limiter-4	400	52000
limiter-5	400	52000

Rate limiters are defined with the `class-of-service application-traffic-control` configuration command.

```
[edit class-of-service]
user@host# set application-traffic-control rate-limiters rate-limiter-name bandwidth-limit value-
in-Kbps burst-rate-limit value-in-bytes
```

Rate-Limiter Assignment

Rate limiters are applied in rules based on the application of the traffic. Two rate limiters are applied for each session: `client-to-server` and `server-to-client`. This usage allows traffic in each direction to be provisioned separately.

The processing of traffic bandwidth by rate limiters is done at the packet level regardless of the direction of traffic. For example: Consider a case where you have only one rate limiter of 10G configured, if the ingress and egress traffic is from the same line card, then the throughput (maximum traffic of both ingress and egress directions combined) can only be up to 10G and not 20G. However, if the device has IOC support (in case of SRX5000 line devices and SRX4600 devices) and ingress traffic is through one IOC and egress traffic through other IOC, then with a single rate-limiter of 10G configured, you can expect a throughput of 20G.

Different AppQoS rules within the same rule set can share a rate limiter. In this case, the applications of those rules share the same bandwidth. There are no limitations on the number of rules in one rule set that can assign the same rate limiter.

The following examples show how the rate limiters defined in the preceding section could be assigned. For instance, a rule set could reuse a rate limiter in several rules and in one or both flow directions:

- rule-set-1
  - rule-1A
    - client-to-server limiter-1
    - server-to-client limiter-1
  - rule-1B
    - client-to-server limiter-1
    - server-to-client limiter-1

If the same profiles are needed in several rule sets, a sufficient number of rate limiters needs to be defined specifying the same `bandwidth-limit` and `burst-size-limit`. The two rule sets in the following example implement the same profiles by assigning different, but comparable, rate limiters.

- rule-set-2

  - rule-2A
    - client-to-server limiter-2
    - server-to-client limiter-2
  - rule-2B
    - client-to-server limiter-2
    - server-to-client limiter-4
- rule-set-3
  - rule-3A
    - client-to-server limiter-3
    - server-to-client limiter-3
  - rule-3B
    - client-to-server limiter-3
    - server-to-client limiter-5

A rate limiter is applied using the `edit class-of-service application-traffic-control rule-sets` command in the same way that a forwarding class, DSCP value, and loss priority are set.

```
[edit class-of-service]
user@host# set application-traffic-control rule-sets rule-set-name rule rule-name1  then rate-
limit client-to-server rate-limiter1 server-to-client rate-limiter2
```

If AppQoS and firewall filter-based rate limiting are both implemented on the egress PIC, both are taken into consideration. AppQoS rate limiting is considered first. Firewall filter-based rate limiting occurs after that.

NOTE: If packets are dropped from a PIC, the device does not send notifications to the client or the server. The upper-level applications on the client and the server devices are responsible for retransmission and error handling.

Rate-Limiter Action

Based on the type of security device, AppQoS rules can be configured with different rate-limiter actions:

- Discard
  - When this option is selected, the out-of-profile packets are just dropped.
  - This is the default action type and need not be configured.
  - This option is supported on all SRX Series Firewalls.
- Loss-priority-high
  - When this option is selected , it elevates the loss priority to maximum. In other words, it is a delayed drop; that is, the discard decision is taken at the egress output queue level. If there is no congestion, it allows the traffic even with maximum loss priority. But if congestion occurs, it drop these maximum loss priority packets first.
  - This option must be configured within the AppQoS rule (to override the default action) using the following command:

    ```
    [edit]
    user@host# set class-of-service application-traffic-control rule-sets rset-01 rule r1 then
    rate-limit loss-priority-high
    ```

  - This option is supported only on for SRX300, SRX320, SRX340, SRX345 devices.

AppQoS Security Policy Configuration

The AppQoS rule set can be implemented in an existing policy or a specific application policy.

```
[edit security policies from-zone zone-name to-zone zone-name]
user@host# set policy policy-name match source-address IP-address
user@host# set policy policy-name match destination-address IP-address
user@host# set policy policy-name match application application-name application-name
user@host# set policy policy-name then permit application-services application-traffic-control
rule-set app-rule-set-name
```

SEE ALSO

Understanding Class of Service

Example: Configuring Application Quality of Service

IN THIS SECTION

This example shows how to enable AppQoS prioritization and rate limiting within a policy.

Requirements

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

Overview

In this example, AppQoS is implemented so that FTP applications are restricted to a level below the specified throughput while other applications are transmitted at a more conventional speed and loss priority level.

Configuration

IN THIS SECTION

Procedure

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, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
set class-of-service forwarding-classes queue 4 my-app-fc
set class-of-service application-traffic-control rate-limiters test-rl bandwidth-limit 100
set class-of-service application-traffic-control rate-limiters test-rl burst-size-limit 13000
set class-of-service application-traffic-control rate-limiters test-r2 bandwidth-limit 200
set class-of-service application-traffic-control rate-limiters test-r2 burst-size-limit 26000
set class-of-service application-traffic-control rule-sets app-test1 rule 0 match application
junos:FTP
set class-of-service application-traffic-control rule-sets app-test1 rule 0 match application
junos:HTTP
set class-of-service application-traffic-control rule-sets app-test1 rule 0 then forwarding-
class my-app-fc
set class-of-service application-traffic-control rule-sets app-test1 rule 0 then dscp-code-point
af22
set class-of-service application-traffic-control rule-sets app-test1 rule 0 then loss-priority
low
set class-of-service application-traffic-control rule-sets app-test1 rule 0 then rate-limit
client-to-server test-r2
set class-of-service application-traffic-control rule-sets app-test1 rule 0 then rate-limit
server-to-client test-r2
set class-of-service application-traffic-control rule-sets app-test1 rule 0 then log
set class-of-service application-traffic-control rule-sets app-test1 rule 1 match application-any
set class-of-service application-traffic-control rule-sets app-test1 rule 1 then rate-limit
client-to-server test-r2
set class-of-service application-traffic-control rule-sets app-test1 rule 1 then rate-limit
server-to-client test-r2
set class-of-service application-traffic-control rule-sets app-test1 rule 1 then log
set security policies from-zone trust to-zone untrust policy p1 match source-address any
set security policies from-zone trust to-zone untrust policy p1 match destination-address any
set security policies from-zone trust to-zone untrust policy p1 match application any
set security policies from-zone trust to-zone untrust policy p1 then permit application-services
application-traffic-control rule-set ftp-test1
```

Step-by-Step Procedure

To configure an AppQoS on your security device:

1. Define one or more forwarding classes dedicated to AppQoS marking. In this example, a single forwarding class, my-app-fc, is defined and assigned to queue 4.

   ```
   [edit]
   user@host# set class-of-service forwarding-classes queue 4 my-app-fc
   ```

For SRX5400, SRX5600, and SRX5800 devices, use the `set class-of-service forwarding-classes class my-app-fc queue 4` command.

Juniper Networks devices support eight queues (0 to 7). The default queues 0 through 3 are assigned to default forwarding classes. Queues 4 through 7 have no default assignments to FCs and are not mapped. To use queues 4 through 7, you must create custom FC names and map them to the queues. For more details, see Forwarding Classes Overview.

2. Define rate limiters. In this example, two rate limiters are defined.

```
[edit]
user@host# set class-of-service application-traffic-control rate-limiters test-rl bandwidth-
limit 100
user@host# set class-of-service application-traffic-control rate-limiters test-rl burst-size-
limit 13000
user@host# set class-of-service application-traffic-control rate-limiters test-r2 bandwidth-
limit 200
user@host# set class-of-service application-traffic-control rate-limiters test-r2 burst-size-
limit 26000
```

For SRX5400, SRX5600, and SRX5800 devices, you can define up to 1000 rate limiters for a device, but only 16 profiles (unique bandwidth-limit and burst-size-limit combinations).

3. Define AppQos rules and application match criteria.

```
[edit]
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 0 match
application junos:FTP
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 0 match
application junos:HTTP
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 0 then
forwarding-class my-app-fc
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 0 then
dscp-code-point af22
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 0 then
loss-priority low
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 0 then
rate-limit client-to-server test-r1
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 0 then
rate-limit server-to-client test-r1
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 0 then
log
```

In this example, when a match is made, the packet is marked with the forwarding class my-app-fc, the DSCP value of af22, and a loss priority of low. We've assigned same rate limiter on both the directions.

You can assign a rate limiter to one or both traffic directions in a single rule. You can also assign a same rate limiter to other rules within a rule set. However, you cannot assign a same rate limiter to different rule set.

4. Define another rule to handle application packets that did not match the previous rule. In this example, a second and final rule applies to all remaining applications.

```
[edit]
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 1 match
application-any
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 1 then
rate-limit client-to-server test-r2
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 1 then
rate-limit server-to-client test-r2
user@host# set class-of-service application-traffic-control rule-sets app-test1 rule 1 then
log
```

5. Add the AppQoS setting to the security policy.

```
[edit]
user@host# set  security policies from-zone trust to-zone untrust policy p1 match source-
address any
user@host# set  security policies from-zone trust to-zone untrust policy p1 match destination-
address any
user@host# set  security policies from-zone trust to-zone untrust policy p1 match application
any
user@host# set  security policies from-zone trust to-zone untrust policy p1 then permit
application-services application-traffic-control rule-set app-test1
```

Results

From configuration mode, confirm your policy configuration by entering the `show security policies` and `show class-of-service` command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

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

```
...
policy p1 {
     match {
        source-address any;
        destination-address any;
        application any;
     }
    then {
        permit {
            application-services {
                application-traffic-control {
                    rule-set app-test1
                }
            }
         }
    }
 }
...
```

```
user@host# show class-of-service
forwarding-classes {
    queue 4 my-app-fc;
}
application-traffic-control {
    rate-limiters test-rl {
        bandwidth-limit 100;
        burst-size-limit 13000;
    }
    rate-limiters test-r2 {
        bandwidth-limit 200;
        burst-size-limit 26000;
    }
    rule-sets app-test1 {
        rule 0 {
            match {
                application [junos:FTP junos:HTTP];
            }
            then {
                forwarding-class my-app-fc;
                dscp-code-point af22;
                loss-priority low;
                rate-limit {
                    client-to-server test-r2;
                    server-to-client test-r2;
                }
                log;
            }
        }
        rule 1 {
            match {
                application-any;
            }
            then {
                rate-limit {
                    client-to-server test-r2;
                    server-to-client test-r2;
                }
                log;
            }
        }
    }
}
```

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

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Flow Session Configuration

Purpose

Verify that AppQoS is enabled.

Action

From operational mode, enter the `show security flow session application-traffic-control extensive` command.

```
user@host> show security flow session application-traffic-control extensive
  Session ID: 3729, Status: Normal, State: Active
  Flag: 0x40
  Policy name: p1
  Source NAT pool: Null
  Dynamic application: junos:FTP
  Application traffic control rule-set: app-test1, Rule: rule0
  Maximum timeout: 300, Current timeout: 276
  Session State: Valid
  Start time: 18292, Duration: 603536
     In: 192.0.2.1/1 --> 203.0.113.0/1;pim,
      Interface: reth1.0,
      Session token: 0x1c0, Flag: 0x0x21
      Route: 0x0, Gateway: 192.0.2.4, Tunnel: 0
      Port sequence: 0, FIN sequence: 0,
      FIN state: 0,
      Pkts: 21043, Bytes: 1136322
     Out: 203.0.113.0/1 --> 192.0.2.0/1;pim,
      Interface: .local..0,
      Session token: 0x80, Flag: 0x0x30
      Route: 0xfffd0000, Gateway: 192.0.2.0, Tunnel: 0
      Port sequence: 0, FIN sequence: 0,
      FIN state: 0,
      Pkts: 0, Bytes: 0
```

Meaning

The entry for application traffic control identifies the rule set and rule of the current session.

Verifying Session Statistics

Purpose

Verify that AppQoS session statistics are being accumulated at each egress node.

Action

From operational mode, enter the `show class-of-service application-traffic-control counter` command.

```
user@host> show class-of-service application-traffic-control counter
pic: 2/1
  Counter type                                          Value
  Sessions processed                                    300
  Sessions marked                                       200
  Sessions honored                                      0
  Sessions rate limited                                 100
  Client-to-server flows rate limited                   100
  Server-to-client flows rate limited                   100

pic: 2/0
  Counter type                                          Value
  Sessions processed                                    400
  Sessions marked                                       300
  Sessions honored                                      0
  Sessions rate limited                                 200
  Client-to-server flows rate limited                   200
  Server-to-client flows rate limited                   200
```

Meaning

The AppQoS statistics are maintained only if application-traffic-control service is enabled. The number of sessions processed, marked, and honored show that sessions are being directed based on configured AppQoS features. The rate-limiting statistics count the number of directional session flows that have been rate limited.

Verifying Rate-Limiter Statistics

Purpose

Verify that bandwidth is being limited as expected when the FTP application is encountered.

Action

From operational mode, enter the `show class-of-service application-traffic-control statistics rate-limiter` command.

```
user@host> show class-of-service application-traffic-control statistics rate-limiter
pic: 2/1
  Ruleset    Application  Client-to-server Rate(kbps)  Server-to-client Rate(kbps)
  app-test1     HTTP      test-r2           200       test-r2          200
  app-test1     HTTP      test-r2           200       test-r2          200
  appp-test1     FTP      test-r1           100       test-r1          100
```

Meaning

Real-time application bandwidth-limit information for each PIC is displayed by rule set. This command provides an indication of the applications being rate limited and the profile being applied.

Verifying Rule Statistics

Purpose

Verify that the rule matches the rule statistics.

Action

From operational mode, enter the `show class-of-service application-traffic-control statistics rule` command.

```
user@host>show class-of-service application-traffic-control statistics rule
pic: 2/1
  Ruleset          Rule           Hits
  app-test1        0              100
  app-test1        1              200
  ...
```

```
pic: 2/0
  Ruleset          Rule           Hits
  app-test1        0              100
  app-test1        1              200
```

Meaning

This command provides information on the number of (session) hits for a rule under each rule set.

SEE ALSO

CoS Device Configuration Overview

Application Quality of Service Support for Unified Policies

IN THIS SECTION

Starting in Junos OS Release 18.2R1, SRX Series Firewalls and vSRX Virtual Firewall instances support unified policies, allowing granular control and enforcement of dynamic Layer 7 applications within the traditional security policy.

Unified policies are the security policies that enable you to use dynamic applications as part of the existing 5-tuple or 6-tuple (5-tuple with a user firewall) match conditions to detect application changes over time.

Application quality of service (AppQoS) is supported when the security device is configured with unified policies. You can configure a default AppQoS rule set to manage unified policy conflicts if multiple security policies match the traffic.

AppQoS rule sets are included in the unified policy to implement application-aware quality-of-service control. You can configure a rule set with rules under the `application-traffic-control` option, and attach the

AppQoS rule set to a unified security policy as an application service. If the traffic matches the specified dynamic application and the policy action is permit, the application-aware quality of service is applied.

Note the following AppQoS functionality in unified policies:

- Upgrading from traditional security policy to a unified policy—In a unified policy, when you configure the `dynamic-application` option as `none`, the AppQoS rule set is applied during the security policy match and the AppQoS looks for the corresponding rule for the identified traffic. This is the same behavior for AppQoS functionality in Junos OS releases prior to Release 18.2R1.

- AppQoS rule with a unified policy—In the application traffic control configuration, the AppQoS rule set is configured with the match condition as `application-any` and in the unified policy, a specific dynamic application is used as the match condition, then, the AppQoS functionality works according to the rule in the unified policy.

Understanding Default Application Quality of Service Rule Set for Unified Policies

You can configure an AppQoS default rule set to manage security policy conflicts.

The initial policy lookup phase occurs prior to identifying a dynamic application. If there are multiple policies present in the potential policy list that contain different AppQoS rule sets, then the security device applies the default AppQoS rule set until a more explicit match has occurred.

You can set an AppQoS as a default AppQoS rule set under the `edit security ngfw` hierarchy level. The default AppQoS rule set is leveraged from one of the existing AppQoS rule sets, which are configured under the `[edit class-of-service application-traffic-control]` hierarchy level.

Table 9 on page 209 summarizes the usage of the default AppQoS rule set under different scenarios in a unified policy.

Table 9: AppQoS Rule Set Usage in Unified Policies

Application Identification Status	AppQoS Rule Set Usage	Action
No security policy conflict.	The AppQoS rule set under the `[edit class-of-service application-traffic-control]` hierarchy is applied when the traffic matches the security policy.	AppQoS is applied as in the AppQoS rule set.

Table 9: AppQoS Rule Set Usage in Unified Policies *(Continued)*

Application Identification Status	AppQoS Rule Set Usage	Action
Security policy conflict and conflicting polices have distinct AppQoS rule sets.	The default AppQoS rule set is not configured or is not found.	Session is ignored because the default AppQoS profile is not configured. As a result, even if the final matched policy in the policy conflict scenario has an AppQoS rule set, this rule set is not applied. We recommend configuring a default AppQoS rule set to manage security policy conflicts.
	The default AppQoS rule set is configured.	AppQoS is applied as in the default AppQoS rule set.
Final application is identified	The matching security policy has an AppQoS rule set, which is same as the default AppQoS rule set.	AppQoS is applied as in the default AppQoS rule set.
	The matching security policy does not have an AppQoS rule set.	Default AppQoS rule set is not applied and AppQoS is not applied for the session.
	The Matching security policy has an AppQoS rule set different from the default AppQoS rule set, which is already applied.	Default AppQoS rule set remains as the default AppQoS rule set.

When a default AppQoS rule set is applied on the traffic and the final security policy has a different AppQoS rule set, in such cases switching from the default AppQoS rule set to the AppQoS rule set in the final security policy is not supported.

Default Application Quality of Service Rule Set In Different Scenarios

The following links are to examples that discuss the default AppQoS rule sets in different scenarios:

Table 10 on page 211 shows different AppQoS rule sets that are configured for unified policies with dynamic applications as the match condition.

Table 10: Different AppQoS Rule Sets in Unified Policies

Security Policy	Source Zone	Source IP Address	Destination Zone	Destination IP Address	Port Number	Protocol	Dynamic Application	Service	AppQoS Rule Set
Policy-P1	S1	50.1.1.1	D1	Any	Any	Any	Facebook	AppQoS	AppQoS-1
Policy-P2	S1	50.1.1.1	D1	Any	Any	Any	Google	AppQoS	AppQoS-2
Policy-P3	S1	50.1.1.1	D1	Any	Any	Any	YouTube	AppQoS	AppQoS-3

In this example, any AppQoS rule sets (AppQoS-1, AppQoS-2, AppQoS-3) can be configured as a default AppQoS rule set under the `[security ngfw]` hierarchy level. It is not necessary for a default rule set to be part of a security policy configuration. Any AppQoS rule set under the `[edit class-of-service application-traffic-control]` hierarchy level can be assigned as the default AppQoS rule set.

No Policy Conflict—All Policies Have the Same AppQoS Rule Set

All matching policies have the same AppQoS rule set as shown in Table 11 on page 211.

Table 11: All Matching Policies Have Same AppQoS Rule Sets

Security Policy	Source Zone	Source IP Address	Destination Zone	Destination IP Address	Port Number	Protocol	Dynamic Application	Service	AppQoS Rule Set
Policy-P1	S1	Any	D1	Any	Any	Any	Facebook	AppQoS	AppQoS-1
Policy-P2	S1	Any	D1	Any	Any	Any	Google	AppQoS	AppQoS-1

In this scenario, the policies Policy-P1 and Policy-P2 have the same AppQoS rule set; that is, AppQoS-1. The rule set AppQoS-1 is applied. Policy-P3 is not configured in this scenario.

If you have configured the rule set AppQoS-2 as the default rule set, it is not applied. That's because there is no conflict in the AppQoS rule sets in the conflicted policies (Policy-P1 and Policy-P2).

No Policy Conflict—All Policies Have the Same AppQoS Rule Set and the Final Policy Has No AppQoS Rule Set

All matching policies have the same AppQoS rule set as shown in Table 12 on page 212 and the final policy has no AppQoS rule set.

Table 12: All Matching Policies Have Same AppQoS Rule Sets and the Final Policy Has No AppQoS Rule Set

Security Policy	Source Zone	Source IP Address	Destination Zone	Destination IP Address	Port Number	Protocol	Dynamic Application	Service	AppQoS Rule Set
Policy-P1	S1	Any	D1	Any	Any	Any	Facebook	AppQoS	AppQoS-1
Policy-P2	S1	Any	D1	Any	Any	Any	Google	AppQoS	AppQoS-1
Policy-P3	S1	50.1.1.1	D1	Any	Any	Any	YouTube	Other	None

In this scenario, both Policy-P1 and Policy-P2 have the same AppQoS rule set, that is, AppQoS-1. In this case, the rule set AppQoS-1 is applied.

When the final policy Policy-P3 is matched, AppQoS ignores the session, because the AppQoS rule set is not configured for Policy-P3.

If the final security policy does not have any AppQoS rule set, then AppQoS is not applied on the traffic. All AppQoS settings that are applied in the prematch stage are reverted to the original values.

Policy Conflict—No AppQoS Rule Set is Configured for the Final Policy

The default AppQoS rule set (in this scenario AppQoS-1) is applied during the potential policy match as shown in Table 13 on page 213. The final policy Policy-P3 has no AppQoS rule set.

Table 13: Matching Policies Have Different AppQoS Rule Sets and the Final Policy Has No AppQoS Rule Set

Security Policy	Source Zone	Source IP Address	Destination Zone	Destination IP Address	Port Number	Protocol	Dynamic Application	Service	AppQoS Rule Set
Policy-P1	S1	50.1.1.1	D1	Any	Any	Any	Facebook	AppQoS	AppQoS-1
Policy-P2	S1	50.1.1.1	D1	Any	Any	Any	Google	AppQoS	AppQoS-2
Policy-P3	S1	50.1.1.1	D1	Any	Any	Any	YouTube	Other	NA

AppQoS ignores the session if the final matching policy Policy-P3 is applied.

If the final security policy does not have any AppQoS rule set, then AppQoS is not applied on the traffic. In this case, all AppQoS settings that are applied in the prematch stage are reverted to the original values.

Policy Conflict—Default AppQoS Rule Set and a Different AppQoS Rule Set for the Final Policy

The rule set AppQoS-1 is configured as a default rule set and is applied when the final application is not yet identified. The final policy Policy-P3 has a different AppQoS rule set (AppQoS-3) as shown in Table 14 on page 213.

Table 14: Different AppQoS Rule Set for the Final Policy

Security Policy	Source Zone	Source IP Address	Destination Zone	Destination IP Address	Port Number	Protocol	Dynamic Application	Service	AppQoS Rule Set
Policy-P1	S1	50.1.1.1	D1	Any	Any	Any	Facebook	AppQoS	AppQoS-1

Table 14: Different AppQoS Rule Set for the Final Policy *(Continued)*

Security Policy	Source Zone	Source IP Address	Destination Zone	Destination IP Address	Port Number	Protocol	Dynamic Application	Service	AppQoS Rule Set
Policy-P2	S1	50.1.1.1	D1	Any	Any	Any	Google	AppQoS	AppQoS-2
Policy-P3	S1	50.1.1.1	D1	Any	Any	Any	YouTube	AppQoS	AppQoS-3

When the final application is identified, the policy Policy-P3 is matched and applied. In this case, the rule set AppQoS-3 is not applied. Instead the rule set AppQoS-1 is applied as the default rule set and remains as the default rule set.

Limitation of AppQoS with Unified Policies

When a security policy is applied to the matching traffic, the AppQoS rule set is applied to the permitted traffic. If the security policy and the applied AppQoS rule set have different dynamic applications, then a conflict might occur as shown in the following example:

```
user@host#  set class-of-service application-traffic-control rule-sets AQ2 rule 1 match
application junos:GOOGLE
user@host#  set class-of-service application-traffic-control rule-sets AQ2 rule 1 then forwarding-
class network-control
user@host#  set class-of-service application-traffic-control rule-sets AQ2 rule 1 then dscp-code-
point 110001
user@host#  set class-of-service application-traffic-control rule-sets AQ2 rule 1 then loss-
priority high
```

```
user@host#  set security policies from-zone trust to-zone untrust policy 1 match source-address
any
user@host#  set security policies from-zone trust to-zone untrust policy 1 match destination-
address any
user@host#  set security policies from-zone trust to-zone untrust policy 1 match application any
user@host#  set security policies from-zone trust to-zone untrust policy 1 match dynamic-
application junos:FTP
```

```
user@host#  set security policies from-zone trust to-zone untrust policy 1  then permit
application-services application-traffic-control rule-set AQ2
```

In this example, the application traffic control rule is configured for junos:GOOGLE and the security policy match condition for the dynamic application is junos: FTP. In such cases, conflicts might occur when the final policy is applied.

SEE ALSO

Application Identification Support for Unified Policies | 95

Example: Configuring Application Quality of Service with Unified Policy

IN THIS SECTION

This example shows how to enable application quality of service (AppQoS) within a unified policy to provide prioritization and rate limiting for the traffic.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall running Junos OS Release 18.2R1 and later. This configuration example is tested for Junos OS Release 18.2R1.

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

Overview

In this example, you configure an AppQoS rule set and invoke AppQoS as an application service in the security policy for the Facebook application.

You define a default AppQoS rule set under the [edit security ngfw] hierarchy level to manage security policy conflicts, if any.

Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

To configure AppQoS with a unified policy:

1. Define an AppQoS rule set.

```
[edit]
user@host# set class-of-service application-traffic-control rule-sets RS1 rule 1 match
application junos:FACEBOOK-APP
user@host# set class-of-service application-traffic-control rule-sets RS1 rule 1 then
forwarding-class fc-appqos loss-priority medium-low dscp-code-point 101110 log
user@host# set class-of-service application-traffic-control rule-sets RS1 rule 1 then rate-
limit client-to-server Ratelimit1
user@host# set class-of-service application-traffic-control rate-limiters Ratelimit1
bandwidth-limit 1000
```

2. Configure a default AppQoS rule set. Select the rule set RS1 that is created under the application traffic control as the default AppQoS rule set.

```
[edit]
user@host# set security ngfw default-profile application-traffic-control rule-set RS1
```

3. Associate the class-of-service rule set to the unified policy.

```
[edit]
user@host# set security policies from-zone untrust to-zone trust policy from_internet match
```

```
source-address any
user@host# set security policies from-zone untrust to-zone trust policy from_internet match
destination-address any
user@host# set security policies from-zone untrust to-zone trust policy from_internet match
application any
user@host# set security policies from-zone untrust to-zone trust policy from_internet match
dynamic-application junos:FACEBOOK-APP
user@host# set security policies from-zone untrust to-zone trust policy from_internet then
permit application-services application-traffic-control rule-set RS1
```

Results

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

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

```
...
policies {
    from-zone trust to-zone untrust {
        policy permit-all {
            match {
                source-address any;
                destination-address any;
                application any;
                dynamic-application junos:FACEBOOK-APP;
            }
            then {
                permit {
                    application-services {
                        application-traffic-control {
                            rule-set RS1;
                        }
                    }
                }
            }
        }
    }
}
...
```

```
ngfw {
    default-profile {
        application-traffic-control {
            rule-set RS1;
        }
    }
}
```

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

Verification

IN THIS SECTION

Confirm that the configuration is working properly.

Verifying Flow Session Configuration

Purpose

Display AppQoS session statistics.

Action

From operational mode, enter the `show class-of-service application-traffic-control counter` command.

Sample Output

command-name

```
pic: 0/0
  Counter type                                                    Value
 Sessions processed                                               2
 Sessions marked                                                  1
 Sessions honored                                                 1
 Sessions rate limited                                            1
 Client-to-server flows rate limited                              0
 Server-to-client flows rate limited                              1
 Session default ruleset hit                                                      1
 Session ignored no default ruleset                               1
```

Meaning

The output displays the number of sessions processed, marked, and honored. The rate-limiting statistics count the number of directional session flows that have been rate limited.

Verifying Rule Statistics

Purpose

Display the AppQoS rule statistics.

Action

From operational mode, enter the `show class-of-service application-traffic-control statistics rule` command.

```
user@host>show class-of-service application-traffic-control statistics rule

pic: 0/0
 Ruleset                       Rule                              Hits
 RS1                           1                                 1
```

Meaning

The output provides information on the number of sessions matched for the rule under each AppQoS rule set.

SEE ALSO

> *ngfw*

RELATED DOCUMENTATION

Application Identification | 5

Application Firewall | 130

Application Tracking | 166

Advanced Policy-Based Routing | 220

Advanced Policy-Based Routing

IN THIS SECTION

Advanced policy-based routing (APBR) also known as application-based routing, a new addition to Juniper Networks suite, provides the ability to forward traffic based on applications. For more information, see the following topics:

Understanding Advanced Policy-Based Routing

IN THIS SECTION

Advanced Policy-Based Routing

The relentless growth of voice, data, and video traffic and applications traversing on the network requires that networks recognize traffic types to effectively prioritize, segregate, and route traffic without compromising performance or availability.

Starting with Junos OS Release 15.1X49-D60, SRX Series Firewalls support advanced policy-based routing (APBR) to address these challenges.

Advanced policy-based routing is a type of session-based, application-aware routing. This mechanism combines the policy-based routing and application-aware traffic management solution. APBR implies classifying the flows based on applications' attributes and applying filters based on these attributes to redirect the traffic. The flow-classifying mechanism is based on packets representing the application in use.

APBR implements:

- Deep packet inspection and pattern-matching capabilities of AppID to identify application traffic or a user session within an application

- Lookup in ASC for application type and the corresponding destination IP address, destination port, protocol type, and service for a matching rule

If a matching rule is found, the traffic is directed to an appropriate route and the corresponding interface or device.

Benefits of APBR

- Enables you to define the routing behavior based on applications.

- Provides more flexible traffic-handling capabilities and offers granular control for forwarding packets based on application attributes.

Understanding How APBR Works

Lets understand about the APBR components before we discuss working of APBR:

- Create an APBR profile (also referred to as an application profile in this document). The profile includes multiple APBR rules. Each rule includes multiple applications or application groups as match criteria. If the traffic matches any of the application or application groups of a rule, the rule is considered as a match and the profile directs the application traffic to the associated routing-instance.

- APBR profile associates a routing instance with the APBR rule. When the traffic matches an application profile, the associated static route and next hop defined in the routing instance is used to route the traffic for the particular session.

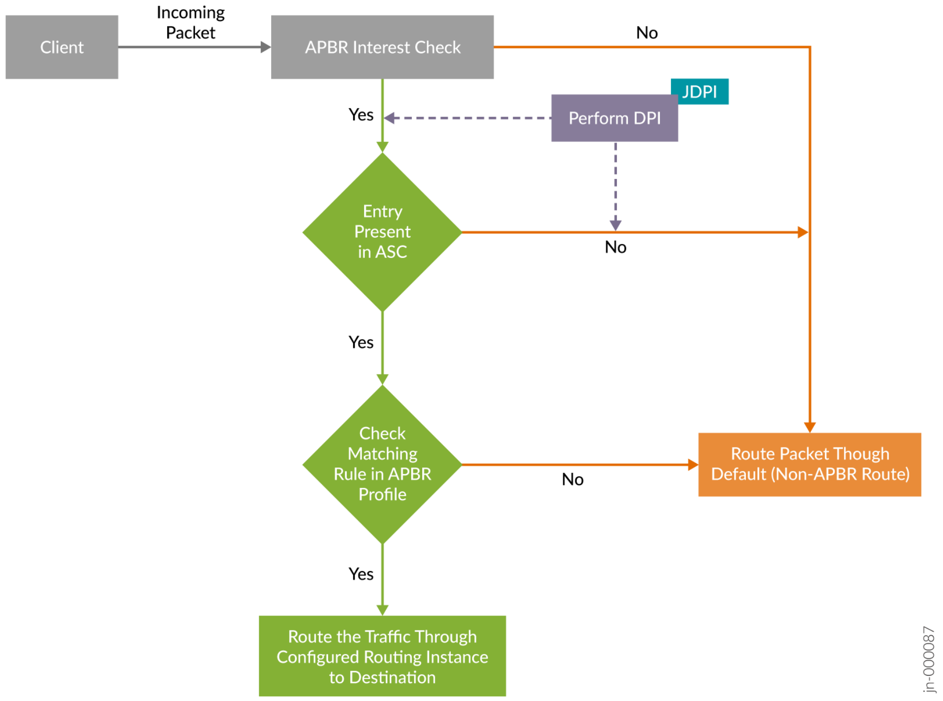
- Associate the application profile to the ingress traffic. You can attach APBR profile to an APBR policy and apply as application services for the session.

Lets proceed with understanding about APBR workflow and then discuss about the APBR midstream support and then about the first packet classification in APBR.

APBR Workflow

Figure 8 on page 223 summarizes APBR behavior prior to Junos OS Release 21.3R1.

Figure 8: APBR Behavior


Client
Incoming Packet
APBR Interest Check
No
JDPI
Perform DPI
Yes
Entry Present in ASC
No
Yes
Check Matching Rule in APBR Profile
No
Route Packet Though Default (Non-APBR Route)
Yes
Route the Traffic Through Configured Routing Instance to Destination
jn-000087

Security device uses DPI to identify attributes of an application and uses APBR to route the traffic over the network. In a service chain, application traffic undergoes DPI before the device applies ABPR. The process of identifying an application using DPI requires analysis of multiple packets. In such cases, initial traffic traverses through a default route (non-APBR route) to reach the destination. The process continues still DPI identifies the application. Once DPI identifies the application, APBR applies rules for the rest of the session. Traffic traverse through the route according to the APBR profile rule.

When you use different NAT pools for source NAT and apply midstream APBR, the source IP address of the session remains same as the one with which the session was using before midstream APBR.

APBR Midstream Support

Starting with Junos OS Release 15.1X49-D110 and Junos OS Release 17.4R1, SRX Series Firewalls support APBR in the middle of a session (which is also known as midstream support). With this enhancement, you can apply APBR for a non-cacheable application and also for the first session of the cacheable application. The enhancement provides more flexible traffic-handling capabilities that offer granular control for forwarding packets.

First packet of session goes through midstream re-routing case. That is, when the application is not yet identified, the traffic traverses through a default route (non-APBR route) to the destination. At the same time, DPI continues till application is identified. Once the application is identified, the device applies APBR profile and the rest of the session packets passes through the route as per the rules defined in the APBR profile. The traffic traverses through a non-APBR route till application signatures or ALG identify the application.

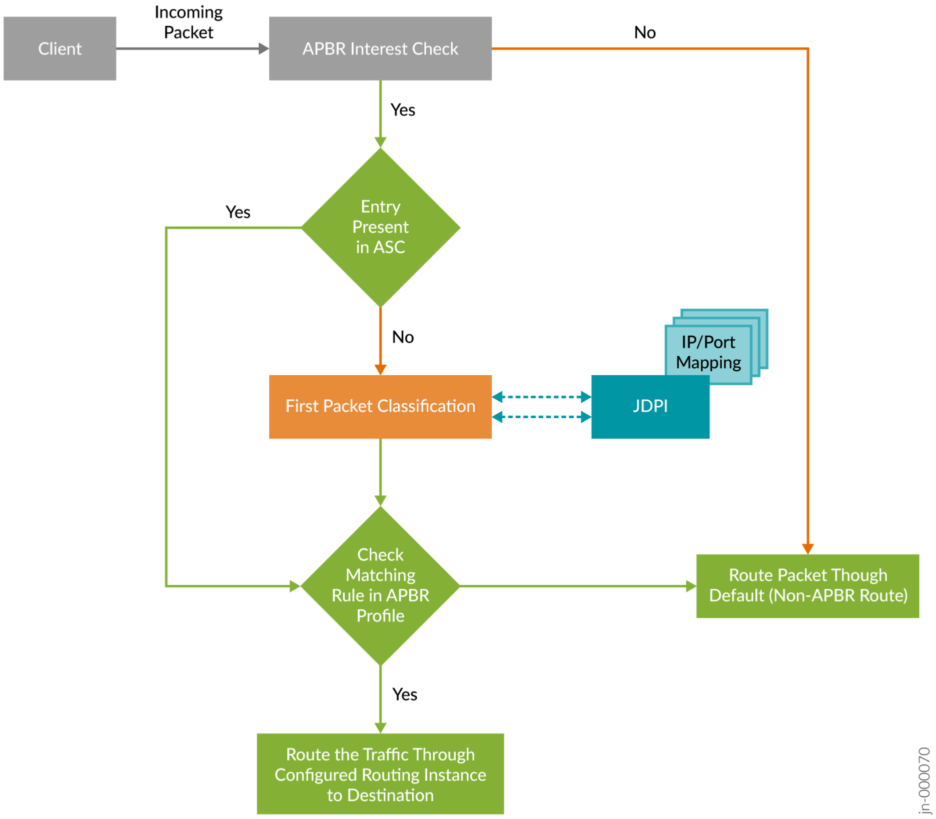
When you use different NAT pools for source NAT and apply midstream APBR, the source IP address of the session remains same as the one with which the session was using before midstream APBR.

APBR with First Packet Classification

Starting in Junos OS Release 21.3R1, APBR uses first-packet classification to identify applications in network traffic. APBR identifies applications by examining the very first packet in the traffic flow and then applies application-specific rules to forward the traffic.

Figure 9 on page 225 shows how APBR uses the first-packet classification to get application details.

Figure 9: APBR with First-Packet Classification


Client
Incoming Packet
APBR Interest Check
No
Yes
Entry Present in ASC
Yes
No
IP/Port Mapping
First Packet Classification
JDPI
Check Matching Rule in APBR Profile
Route Packet Though Default (Non-APBR Route)
Yes
Route the Traffic Through Configured Routing Instance to Destination
jn-000070

First-packet classification leverages on the repository that includes details such as static IP mapping and ports details of applications. The repository is part of the application signature package (JDPI).

For the first session of a cacheable application, APBR queries the ASC to get application details of the flow. If the entry of the application is not available in the ASC, APBR queries JDPI for the application details. APBR uses the IP address and the port details of the flow for the query. If the application mapping is available, then JDPI returns the details to APBR. After getting application details, APBR searches for the configured profile of the application and routes the packet through the assigned routing instance.

At the same time, JDPI continues to process the packets and updates the ASC (if enabled). For the subsequent flows, APBR performs routing of the traffic based on the application entry present in the ASC for the flow.

With first-packet classification, you can use different NAT pools for source NAT in your APBR configuration for cacheable application.

Benefits of First Packet Classification

With first-packet classification, you can steer the traffic accurately and efficiently over the network, optimizing network link utilization and boosting the performance.

Limitations

- For non-cacheable applications, when you use different NAT pools for the source NAT and apply APBR in the middle of the session, the source IP address of the session continues to remain same even after applying the APBR in the midstream.

- If IP address and port range details of an application changes, the change might not immediately reflect in the application signature package. You must install the application signature package to get the latest updates of IP address and port ranges.

- In case of micro-services hosting multiple applications such as OFFICE365 on the cloud, it is not possible to have IP addresses and ports ranges at a granular level. For such cases, the first packet classification returns the parent application details. You must configure APBR profile rule to include nested application and the parent application. Example: create APBR rule with dynamic application as MS-TEAMS, and add OFFICE365-CREATE-CONVERSATION in the same rule for first packet classification.

Advanced Policy-Based Routing Options

You can streamline the traffic handling with APBR by using the following options:

- **Limit route change-** Some sessions go through continuous classification in the middle of the session as application signatures identify the application. Whenever an application is identified by the application signatures, APBR is applied, and this results in a change in the route of the traffic. You can limit the number of times a route can change for a session by using the `max-route-change` option of the `tunables` statement.

  ```
  set security advance-policy-based-routing tunables max-route-change value
  ```

  Example:

  ```
  [edit]
  set security advance-policy-based-routing tunables max-route-change 5
  ```

  In this example, you want to limit the number of route changes per session to 5. When there is a change in the route in the middle of the session, this count is reduced to 4. This process continues until the count reaches 0. After that, APBR is not applied in the middle of the session.

If an identified application has an entry in the ASC, then, the count is not reduced for that session, because the session started with the specified route according to the APBR configuration.

- **Terminate session if APBR is bypassed**–You can terminate the session if there is a mismatch between zones when APBR is being applied in the middle of the session. When you want to apply APBR in the middle of a session, both new egress interface and existing egress interface must be part of the same zone. If you change the zone for an interface in the middle of a session, then, by default, APBR is not applied, and the traffic continues to traverse through the existing interface. To change this default behavior, you can terminate the session entirely, instead of allowing traffic to traverse through the same route bypassing APBR, by using the `drop-on-zone-mismatch` option of the `tunables` statement.

  Example:

  ```
  [edit]
  set security advance-policy-based-routing tunables drop-on-zone-mismatch
  ```

- **Enable logging**—You can enable logging to record events that occur on the device, for instance, when APBR is bypassed because of a change in the zones for interfaces. You can use the `enable-logging` option of the `tunables` statement to configure the logging.

  Example:

  ```
  [edit]
  set security advance-policy-based-routing tunables enable-logging
  ```

- **Enable reverse reroute**—For deployments that requires traffic symmetry for ECMP routes, and the incoming traffic needs to switch in the middle of session, the rerouting can be achieved using the option enable-reverse-reroute specific to a security zone as follows:

  Example:

  `[edit]`

  `set security zones security-zone zone-name enable-reverse-reroute`

  When the above configuration is enabled for a security zone, where an incoming packet arrives on an interface and has a different outgoing/return interface, a change in the interface is detected and triggers a reroute. A route lookup is performed for the reverse path, and the preference will be given to the interface on which the packet has arrived.

  Further processing stops for a particular session when a route lookup fails for the traffic on reverse path.

Support for reverse rerouting is available starting in Junos OS Release 15.1X49-D130 and later releases.

- **Support for Layer 3 and Layer 4 Applications**—Starting in Junos OS Release 20.2R1, APBR supports Layer 3 and Layer 4 custom applications. You can manually disable Layer 3 and Layer 4 custom application lookup in APBR by using the following configuration-statement:

```
user@host# set security advance-policy-based-routing tunables no-l3l4-app-lookup
```

- **Application Tracking**—
  You can enable, AppTrack to inspect traffic and collect statistics for application flows in the specified zone. See "Understanding Application Tracking" on page 167 for more details.

Use Case

- When multiple ISP links are used:
  - APBR can be used for selecting high-bandwidth, low-latency links for important applications, when more than one link is available.
  - APBR can be used for creating a fallback link for important traffic in case of link failure. When multiple links are available, and the main link carrying the important application traffic suffers an outage, then the other link configured as the fallback link can be used to carry traffic.
  - APBR can be used for segregating the traffic for deep inspection or analysis. With this feature, you can classify the traffic based on applications that are required to go through deep inspection and audit. If required, such traffic can be routed to a different device.

Limitations

APBR has the following limitations:

- Redirecting the route for the traffic depends on the presence of an entry in the application system cache (ASC). Routing will succeed only if the ASC lookup is successful. For the first session, when the ASC is not present for the traffic, the traffic traverses through a default route (non-APBR route) to the destination (this limitation is applicable only for the releases before Junos OS 15.1X49-D110).
- APBR does not work if an application signature package is not installed or application identification is not enabled.

APBR with midstream support has the following limitations:

- APBR works only for forward traffic.

- APBR does not work for data sessions initiated by an entity from the control session, such as active FTP.

- When using different NAT pools for source NAT and midstream APBR is applied, the source IP address of the session continues to be the same as the one with which the session has been using before applying the midstream APBR.

- APBR with midstream support works only when all egress interfaces are in the same zone. Because of this, only the forwarding and virtual routing and forwarding (VRF) routing instances can be used to avail APBR midstream support.

SEE ALSO

Example: Configuring Advanced Policy-Based Routing for Application-Aware Traffic Management Solution | 229

Example: Configuring Advanced Policy-Based Routing for Application-Aware Traffic Management Solution

IN THIS SECTION

This example shows how to configure APBR on an SRX Series Firewall.

Requirements

This example uses the following hardware and software components:

- Valid application identification feature license installed on an SRX Series Firewall.

- SRX Series Firewall with Junos OS Release 15.1X49-D60 or later. This configuration example is tested for Junos OS Release 15.1X49-D60.

Overview

In this example, you want to forward HTTP, social networking, and Yahoo traffic arriving at the trust zone to a specific device or interface as specified by the next-hop IP address.

When traffic arrives at the trust zone, it is matched by the APBR profile, and if a matching rule is found, the packets are forwarded to the static route and next hop as specified in the routing instance. The static route configured in the routing table is inserted into the forwarding table when the next-hop address is reachable. All traffic destined for the static route is transmitted to the next-hop address for transit to a specific device or interface.

Figure 10 on page 230 shows the topology used in this configuration example.

Figure 10: Topology For Advanced Policy-Based Routing (APBR)

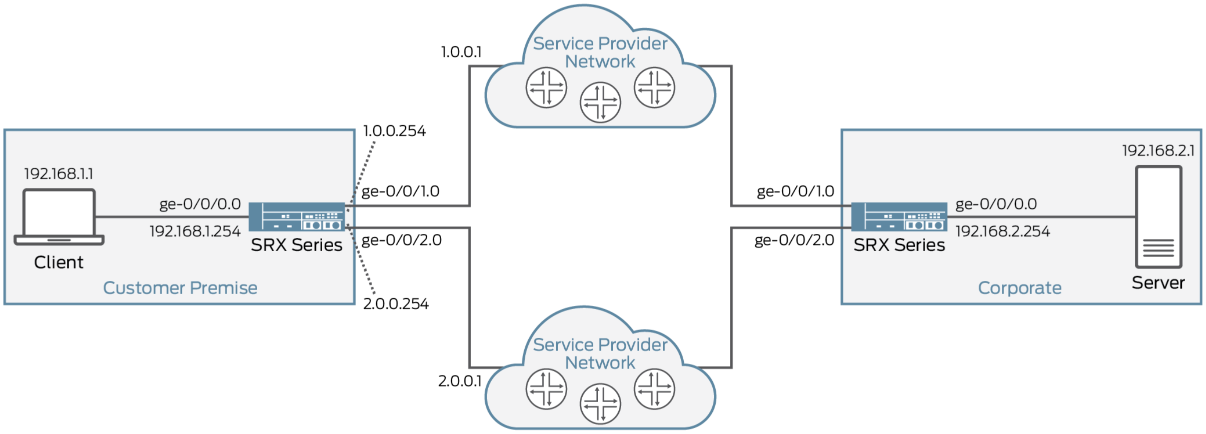


Table 15 on page 230 provides the details of the parameters used in this example.

Table 15: APBR Configuration Parameters

Parameter	Name	Description
Routing Instance	• Instance name—R1 • Instance type— forwarding • Static route— 192.168.0.0/16 • Next-hop— 1.0.0.1	Routing instance of type forwarding is used for forwarding the traffic. All the qualified traffic destined for the static route (example: 192.168.0.0/16) is forwarded to the next-hop device (example: with 1.0.0.1 address on its interface).

Table 15: APBR Configuration Parameters *(Continued)*

Parameter	Name	Description
	• Instance name—R2 • Instance type— forwarding • Static route—192.168.0.0/16 • Next-hop— 2.0.0.1	
RIB Group	apbr_group	Name of the routing information base (RIB) (also known as routing table) group. This RIB group is configured to import interface route entries from inet.0, RI1.inet.0, RI2.inet.0, and RI3.inet.0.
APBR Profile	profile-1	Name of the APBR profile. This profile matches applications and application groups and redirects the matching traffic to the specified routing instance (example: R1) for the route lookup. The profile includes multiple rules.
Rule	• Rule name—ruleApp1 • matching application—junos:HTTP • Associated routing instance—R1	Define the rules for the APBR profile. Associate the rule with one or more than one application (example: for HTTP) or application groups. If the application matches any of the application or application groups of a rule in a profile, the application profile rule is considered as a match and the traffic will be redirected to the routing instance (example: R1) for the route lookup.
	• rule name—ruleApp2 • matching application—junos:web:social-networking • Routing instance— R2	
Zone	trust	Specify the source zone to which the APBR profile can be applied.

NOTE: To use the APBR for redirecting the traffic based on applications, importing interface routes might be required from one routing instance to another routing instance. You can use one of the following mechanisms:

- RIB groups to import interface routes
- Routing policy to import interface routes

When you use routing policy to import interface routes, it might cause management local routes (using fxp0) to leak to non-default routing instance, if the appropriate action is not used for the routing policy. When devices are in chassis cluster mode, such scenarios might result in RG0 failover due to limitations. We recommend not configure fxp0 local route in the routing table of non-default routing instance. Following sample depicts a sample configuration of policy options. Note that the reject action helps in eliminating the routes that are not required. You can use specific routes to reject the fxp0 routes.

```
policy-statement statement-name {
    term 1 {
        from {
            instance master;
            route-filter route-filter-ip-address exact;
        }
        then accept;
    }
    then reject;
}
```

NOTE: APBR is used for routing the packets in a forward path. For return traffic to arrive over the same path, we recommend to configure the remote SRX Series Firewall with ECMP configuration along with load balance routing policy as shown in the following sample configuration:

```
user@host> set routing-options static route ip-address next-hop ip-address
user@host> set routing-options static route ip-address next-hop ip-address
user@host> set policy-options policy-statement load-balance-policy then load-balance per-packet
user@host> set routing-options forwarding-table export load-balance-policy
```

Configuration

IN THIS SECTION

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 routing-instances R1 instance-type forwarding
set routing-instances R1 routing-options static route 192.168.0.0/16 next-hop 1.0.0.1
set routing-instances R2 instance-type forwarding
set routing-instances R2 routing-options static route 192.168.0.0/16 next-hop 2.0.0.1
set routing-options interface-routes rib-group inet apbr_group
set routing-options rib-groups apbr_group import-rib inet.0
set routing-options rib-groups apbr_group import-rib RI1.inet.0
set routing-options rib-groups apbr_group import-rib RI2.inet.0
set security advance-policy-based-routing profile profile1 rule rule-app1 match dynamic-
application junos:HTTP
set security advance-policy-based-routing profile profile1 rule rule-app1 then routing-instance
R1
set security advance-policy-based-routing profile profile1 rule rule-app2 match dynamic-
application-group junos:web:social-networking
set security advance-policy-based-routing profile profile1 rule rule-app2 then routing-instance
R2
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 ge-0/0/0.0
set security zones security-zone trust advance-policy-based-routing-profile profile1
```

Configuring Advanced Policy-Based Routing

Step-by-Step Procedure

To configure APBR:

1. Create routing instances.

```
[edit]
user@host# set routing-instances R1 instance-type forwarding
user@host# set routing-instances R1 routing-options static route 192.168.0.0/16 next-hop
1.0.0.1
user@host# set routing-instances R2 instance-type forwarding
user@host# set routing-instances R2 routing-options static route 192.168.0.0/16 next-hop
2.0.0.1
```

2. Group one or more routing tables to form a RIB group called apbr_group and import routes into the routing tables.

```
[edit]
set routing-options interface-routes rib-group inet apbr_group
set routing-options rib-groups apbr_group import-rib inet.0
set routing-options rib-groups apbr_group import-rib RI1.inet.0
set routing-options rib-groups apbr_group import-rib RI2.inet.0
```

3. Create the APBR profile and define the rules.

```
[edit]
user@host# set security advance-policy-based-routing profile profile1 rule rule-app1 match
dynamic-application junos:HTTP
user@host# set security advance-policy-based-routing profile profile1 rule rule-app1 then
routing-instance R1
user@host# set security advance-policy-based-routing profile profile1 rule rule-app2 match
dynamic-application-group junos:web:social-networking
user@host# set security advance-policy-based-routing profile profile1 rule rule-app2 then
routing-instance R2
```

4. Apply the APBR profile to the security zone.

```
[edit]
user@host# set security zones security-zone trust host-inbound-traffic system-services all
user@host# set security zones security-zone trust host-inbound-traffic protocols all
user@host# set security zones security-zone trust interfaces ge-0/0/0.0
user@host# set security zones security-zone trust advance-policy-based-routing-profile
profile1
```

Results

From configuration mode, confirm your configuration by entering the `show routing-instances` 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 routing-instances
R1 {
    instance-type forwarding;
    routing-options {
        static {
            route 192.168.0.0/16 next-hop 1.0.0.1;
        }
    }
}
R2 {
    instance-type forwarding;
    routing-options {
        static {
            route 192.168.0.0/16 next-hop 2.0.0.1;
        }
    }
}
```

```
[edit]
user@host# show routing-options
interface-routes {
    rib-group inet apbr_group;
}
```

```
rib-groups {
    apbr_group {
        import-rib [ inet.0 RI1.inet.0 RI2.inet.0 ];
    }
}
```

```
[edit]
user@host# show security advance-policy-based-routing
profile profile1 {
    rule rule-app1 {
        match {
            dynamic-application junos:HTTP;
        }
        then {
            routing-instance R1;
        }
    }
    rule rule-app2 {
        match {
            dynamic-application-group junos:web:social-networking;
        }
        then {
            routing-instance R2;
        }
    }
}
```

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

```
            }
    advance-policy-based-routing-profile {
        profile1;
    }
}
```

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

Verification

IN THIS SECTION

Verifying Advanced Policy-Based Routing Statistics

Purpose

Display the statistics for APBR such as the number of sessions processed for the application-based routing, number of times the APBR is applied for the session, and so on.

Action

From configuration mode, enter the `show security advance-policy-based-routing statistics` command.

```
user@host>  show security advance-policy-based-routing statistics
Advance Profile Based Routing statistics:
  Sessions Processed                        9
  App rule hit on cache hit                 0
  App rule hit on HTTP Proxy/ALG            0
  Midstream disabled rule hit on cache hit  2
  URL cat rule hit on cache hit             0
  DSCP rule hit on first packet             2
  App and DSCP hit on first packet          0
  App rule hit midstream                    1
  Default rule match                        0
  Midstream disabled rule hit midstream     1
```

```
    URL cat rule hit midstream                0
    App and DSCP rule hit midstream           0
    DSCP rule hit midstream                   0
    Route changed on cache hits               2
    Route changed on HTTP Proxy/ALG           0
    Route changed midstream                   0
    Default rule applied                      0
    Zone mismatch                             0
    Drop on zone mismatch                     0
    Next hop not found                        0
    Application services bypass               0
```

Meaning

The command output displays the following details:

- Sessions processed for the application-based routing.
- The number of times the application traffic matches the APBR profile and APBR is applied for the session based on different criteria.
- The number of times AppID was consulted to identify application traffic.
- The number of instances when there was a mismatch in security zone of the default route and the APBR selected route and the traffic was dropped due to this mismatch.

See *show security advance-policy-based-routing statistics* for more details.

Verifying Advanced Policy-Based Routing

Purpose

Display information about the sessions and packet flows active on the device, including detailed information about specific sessions.

Action

From configuration mode, enter the `show security flow session` command to display information about all currently active security sessions on the device.

Meaning

The command output displays the following details:

- All active sessions and packet flows on your device
- List of incoming and outgoing IP flows, including services
- Security attributes associated with a flow, for example, the policies that apply to traffic belonging to that flow
- Session timeout value, when the session became active, how long the session has been active, and if there is active traffic on the session

SEE ALSO

Understanding Advanced Policy-Based Routing | 221

Configuring Advanced Policy-Based Routing Policies

IN THIS SECTION

Starting in Junos OS Release 18.2R1, you can configure advanced policy-based routing (APBR) policies by defining source addresses, destination addresses, and applications as match conditions; and after a successful match, the configured APBR profile is applied as an application services for the session. In the previous releases of Junos OS, an APBR profile could be attached to an incoming security zone of the ingress traffic, and the APBR was applied per security zone basis. Now, with support of APBR policies, you can apply different set of APBR rules on the traffic based on incoming security zone, source address, destination address and application

This enhancement provides more flexible traffic-handling capabilities that offer granular control for forwarding packets.

Supported match criteria includes source addresses, destination addresses, and applications. The applications can be used to support the matching condition based on protocol and Layer 4 ports.

If one or more APBR policy is configured for the security zone, then the policy is evaluated during session creating phase. The policy lookup is terminated once the policy, matching the session, is selected. After a successful match, the APBR profile configured with the APBR policy is used for the session.

How APBR Policy Works?

APBR policies are defined for a security zone. If there is one or more APBR policy associated with a zone, the session that is initiated from the security zone goes through the policy match.

The following sequences are involved in matching the traffic by an APBR policy and applying advanced policy-based routing to forward the traffic, based on the defined parameters/rules:

- When traffic arrives at the ingress zone, it is matched by the APBR policy rules. The policy match condition include the source address, destination address and application.
- When the traffic matches the security policy rules, the action of the APBR policy is applied to the traffic. You can enable APBR as an application service in the APBR policy action by specifying the APBR profile name.
- The APBR profile configuration incudes the set of rules that contains set of dynamic applications and dyamic application groups as match condition. The action part of those rules contain the routing instance through which traffic needs to be forwarded. The routing instance can include configuration of static routs or dynamic learned routes.
- All traffic destined for the static route is transmitted to the next-hop address for transit to a specific device or interface.

APBR policy rules are terminal, which means that once the traffic is matched by a policy, it is not processed further by the other policies.

If an APBR policy has the matching traffic and APBR profile does not have any traffic matching the rule, then the traffic matching the APBR policy traverses through a default routing-instance [inet0] to the destination.

Legacy APBR Profile Support

Prior to the Junos OS Release 18.2R1, APBR profile was applied at security zone-level. With the support for APBR policy, APBR configuration at security-zone level is deprecated future, rather than being immediately removed in order to provide backward compatibility and a chance to bring your configuration into compliance with the new configuration.

However, if you have configured a zone-based APBR, and you attempt to add an APBR policy for the particular security zone, commit might fail. You must delete the zone-based configuration in order to configure the APBR policy for the zone. Similarly if an APBR policy is configured for a security zone, and you attempt to configure zone-based APBR, results in commit error.

Limitation

- When using specific address or address set in the APBR policy rule, we recommend to use the global address book. Because, zone specific rules might not be applicable for destination address, as the destination zone is not known at time of policy evaluation.
- Configuring APBR policy for the security zone junos-host zone is not supported.

Example: Configuring Advanced Policy-Based Routing Policies

IN THIS SECTION

This example shows how configure an APBR policy and apply the APBR profile on the session that matches the APBR policy rules.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall with Junos OS Release 18.2R1 or later. This configuration example is tested on Junos OS Release 18.2R1.
- Valid application identification feature license installed on an SRX Series Firewall.

Overview

In this example, you want to forward HTTP traffic arriving at the trust zone to a specific device or interface as specified by the next-hop IP address.

When traffic arrives at the trust zone, it is matched by the APBR policy. When the traffic matches the policy, the configured APBR rule is applied on the permitted traffic as application services. The packets are forwarded based on the APBR rule to the static route and next hop as specified in the routing instance. The static route configured in the routing table is inserted into the forwarding table when the next-hop address is reachable. All traffic destined for the static route is transmitted to the next-hop address for transit to a specific device or interface.

In this example, you must complete the following configurations:

- Define routing instance and RIB group.
- Create an ABPR profile.
- Create a security zone.
- Create an APBR policy and attach the APBR profile to it.

Configuration

IN THIS SECTION

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 routing-instances R1 instance-type forwarding
set routing-instances R1 routing-options static route 5.0.0.0/24 next-hop 3.0.0.2
set routing-options interface-routes rib-group inet fbf-group
set routing-options rib-groups fbf-group import-rib inet.0
set routing-options rib-groups fbf-group import-rib RI1.inet.0
set security advance-policy-based-routing profile profile1 rule rule-app1 match dynamic-
application junos:HTTP
set security advance-policy-based-routing profile profile1 rule rule-app1 then routing-instance
R1
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 ge-0/0/1.0
set security advance-policy-based-routing from-zone trust policy SLA1 match source-address any
set security advance-policy-based-routing  from-zone trust policy SLA1 match destination-address
any
set security advance-policy-based-routing  from-zone trust policy SLA1 match application any
set security advance-policy-based-routing  from-zone trust policy SLA1 then application-services
advance-policy-based-routing-profile profile1
```

Configuring Advanced Policy-Based Routing

Step-by-Step Procedure

To apply APBR on the traffic matching the APBR policy:

1. Create routing instances.

   ```
   [edit]
   user@host# set routing-instances R1 instance-type forwarding
   user@host# set routing-instances R1 routing-options static route 5.0.0.0/24 next-hop 3.0.0.2
   ```

2. Group one or more routing tables to form a RIB group called apbr_group and import routes into the routing tables.

   ```
   [edit]
   user@host# set routing-options interface-routes rib-group inet fbf-group
   user@host# set routing-options rib-groups fbf-group import-rib inet.0
   user@host# set routing-options rib-groups fbf-group import-rib RI1.inet.0
   ```

3. Create the APBR profile and define the rules.

   ```
   [edit]
   user@host# set security advance-policy-based-routing profile profile1 rule rule-app1 match
   dynamic-application junos:HTTP
   user@host# set security advance-policy-based-routing profile profile1 rule rule-app1 then
   routing-instance R1
   ```

4. Create a security zone.

   ```
   [edit]
   user@host# set security zones security-zone trust host-inbound-traffic system-services all
   ```

```
user@host# set security zones security-zone trust host-inbound-traffic protocols all
user@host# set security zones security-zone trust interfaces ge-0/0/1.0
```

5. Create an APBR policy and apply the APBR profile to the security zone.

```
[edit]
user@host# set security advance-policy-based-routing from-zone trust policy SLA1 match source-
address any
user@host# set security advance-policy-based-routing from-zone trust policy SLA1 match
destination-address any
user@host# set security advance-policy-based-routing  from-zone trust policy SLA1 match
application any
user@host# set security advance-policy-based-routing  from-zone trust policy SLA1 then
application-services advance-policy-based-routing-profile profile1
```

Results

From configuration mode, confirm your configuration by entering the `show routing-instances` 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 routing-instances
R1 {
    instance-type forwarding;
    routing-options {
        static {
            route 5.0.0.0/24 next-hop 3.0.0.2;
        }
    }
}
```

```
[edit]
user@host# show routing-options
interface-routes {
    rib-group inet fbf_group;
}
rib-groups {
    fbf_group {
        import-rib [ inet.0 RI1.inet.0];
    }
}
```

```
[edit]
user@host# show security advance-policy-based-routing
from-zone trust {
    policy SLA1 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            application-services {
                advanced-policy-based-routing-profile profile1;
            }
        }
    }
}
profile profile1 {
    rule rule-app1 {
        match {
            dynamic-application junos:HTTP;
        }
        then {
            routing-instance R1;
        }
    }
}
```

```
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
```

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

Verification

IN THIS SECTION

Verifying Advanced Policy-Based Routing Statistics

Purpose

Display the statistics for APBR such as the number of sessions processed for the application-based routing, number of times the APBR is applied for the session, and so on.

Action

From configuration mode, enter the `show security advance-policy-based-routing statistics` command.

```

Sessions Processed                     18994
  AppID cache hits                       18994
  AppID requested                        0
  Rule matches                           0
  Route changed on cache hits            0
  Route changed midstream                0
  Zone mismatch                          0
  Drop on zone mismatch                  0
  Next hop not found                     0
```

Meaning

The command output displays the following details:

- Sessions processed for the application-based routing.
- The number of times the application traffic matches the APBR profile and APBR is applied for the session.
- The number of times AppID was consulted to identify application traffic.

See *show security advance-policy-based-routing statistics* for more details.

Verifying APBR Policy Configuration

Purpose

Display information about the APBR policy, associated APBR profile and to display information about the APBR policy hit count.

Action

From configuration mode, enter the `show security advanced-policy-based-routing` command.

```
user@host> show security advanced-policy-based-routing policy-name SLA1

From zone: trust
  Policy: SLA1, State: enabled, Index: 7, Sequence number: 1
    Source addresses: any
    Destination addresses: any
    Applications: any
    APBR profile: profile1
```

From configuration mode, enter the `show security advanced-policy-based-routing hit-count` command.

```
user@host> show security advanced-policy-based-routing hit-count
Logical system: root-logical-system
 Index   From zone        Name           Hit count
 1       trust            SLA1          3
 2       trust            SLA2          0
 3       trust            SLA1          0
```

```
Number of policy: 3
```

Meaning

The command output displays the following details:

- Details such as status of the policy, associated APBR profile.
- Display the utility rate of policies according to the number of hits they receive.

SEE ALSO

Understanding Advanced Policy-Based Routing

Understanding URL Category-Based Routing

IN THIS SECTION

Starting in Junos OS Release 18.3 R1, URL category-based routing is supported on SRX Series Firewalls and vSRX Virtual Firewall instances. URL category-based routing enables you to use URL categories as match criteria in an APBR profile. The URL categories are based on the destination server IP address, and the category identification is leveraged from the Enhanced Web Filtering (EWF) and local Web filtering results obtained from the Content Security module.

URL category-based routing enables you to identify and selectively route Web traffic (HTTP and HTTPS) to a specified destination.

Web filtering classifies websites into the categories according to host, URL, or IP address, and performs the filtering based on those categories. You can configure APBR profiles by specifying a URL category as the match condition in the rule. The APBR profile rule matches the traffic with specified match criteria, and after a successful match, the configured APBR profile is applied as the application service for the session. For example, suppose you want to route all the traffic belonging to a specific website category, such as social media, through a specific next hop. In this case, you can create a new APBR profile with the list of URL categories such as Enhanced_Social_Web_Facebook, Enhanced_Social_Web_Linkedin, Enhanced_Social_Web_Twitter or Enhanced_Social_Web_Youtube or any other custom URL as match criteria in the policy. The traffic that matches one of the defined URL categories in the rule is forwarded using the routes of the specific routing instance.

When an APBR profile matches the traffic against the URL categories included in the rule, APBR queries the Web filtering module to get the URL category details. If the URL category is not available in the URL filtering cache, then the security device sends a request to the private cloud configured with Web filtering for the categorization details. If the traffic does not match any URL categories, the request is uncategorized, and the session undergoes normal processing (non-APBR route).

NOTE: If the private cloud configured with EWF does not respond to the URL category request within an interval of 3 seconds, then the session undergoes normal processing (non-APBR route).

Rule Processing in an APBR Profile

You can provide advanced policy-based routing by classifying the traffic based on applications' attributes and applying policies based on these attributes to redirect the traffic. To do this, you must define the APBR profile and associate it to a APBR policy. You can create an APBR profile to include multiple rules with either dynamic applications, application groups or both, or a URL category as match criteria. The rules configured in the APBR profile can include either of the following:

- One or more applications, dynamic applications, or application groups
- URL category (IP destination address)—EWF or local Web filtering.

In an APBR profile, rule lookup is performed for both the match criteria. If only one match criteria is available, the rule lookup is done based on the available match criteria.

The APBR profile includes the rules to match the traffic with applications or URL categories and the action to redirect the matching traffic to the specified routing instance for the route lookup.

In Junos OS Release18.3R1, the URL category match is done based on the destination IP address; because of this, URL category-based rule match is terminated at the first packet of the session. As a dynamic application might be identified in the middle of the session, the matching process for the dynamic application rules continues until the process of application identification is complete.

Benefits of URL Category-Based Routing

-

Using URL-based categories enables you to have granular control over Web traffic. The traffic belonging to specific categories of websites is redirected through different paths, and based on the category, it is subjected to further security processing, including SSL decryption for HTTPS traffic.

- Traffic-handling capabilities based on URL categories enable you to use different paths for selected websites. Using different paths results in better quality of experience (QoE) and also enables you to utilize the available bandwidth effectively.
- SD-WAN solutions can utilize URL category-based routing in addition to the dynamic application-based routing.
- URL category-based routing can be used for local Internet breakout solutions as it can work with source NAT configuration changes.

Limitations of URL Category-Based Routing

Using URL categories in an APBR profile has the following limitations:

- Only the destination IP address is used for the URL category identification in an APBR profile. URL categories based on the host, or on the URL or the SNI field are not supported.
- You can configure either a dynamic application or a URL category as the match condition in an APBR profile rule. Configuring a rule with both URL category and dynamic application results in a commit error.

Example: Configuring URL Category-Based Routing

IN THIS SECTION

This example shows you how to configure URL category-based routing.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall with Junos OS Release 18.3 R1 or later. This configuration example is tested on Junos OS Release 18.3 R1.
- Valid application identification feature license installed on the SRX Series Firewall.
- The Enhanced Web Filtering (EWF) option requires you to purchase a Juniper Networks Web filtering license. No license is required for local Web filtering.

Overview

This example shows how to configure APBR on your SRX Series Firewall to forward social media traffic arriving at the trust zone to a specific device or to an interface using URL category-based routing.

When traffic arrives, it is matched by the APBR profile, and if a matching rule is found, the packets are forwarded to the static route and next-hop IP address as specified in the routing instance. The static route configured in the routing table is added to the forwarding table when the next-hop address is reachable. All traffic destined for the static route is transmitted to the next-hop address for transit to a specific device or to an interface.

In this example, you complete the following configurations:

- Enable either of the following types of Web filtering:
  - Enhanced Web Filtering (EWF)—When you enable EWF on the device, the EWF engine intercepts the HTTP and the HTTPS requests and categorizes the URL into one of the 95 or more predefined categories and also provides site reputation information. See "Configuring URL-Based Routing by Using Local Web Filtering" on page 257.
  - Local Web filtering—When you enable local Web filtering, you can configure custom URL categories with multiple URL lists and apply them to a Content Security Web filtering profile with actions such as permit, permit and log, block, and quarantine. To use local Web filtering, you must create a Web filtering profile and ensure that category custom is part of the profile. See "Configuring URL Category-Based Routing by Using EWF" on page 252.
- Define the routing instances and the routing information base (RIB; also known as routing table group.)
- Define the APBR profile and associate it to an APBR policy.

Configuring URL Category-Based Routing by Using EWF

IN THIS SECTION

This section provides the steps to configure URL category-based routing using EWF. Table 16 on page 252 provides the details of the parameters used in this example.

Table 16: Configuration Parameters for URL Category-Based Routing Using EWF

Parameters	Name	Description
APBR profile	apbr-pr1	Name of the APBR profile.
APBR policy	p1	Name of the APBR policy.
Rule	• Rule name—rule rule-social-nw • Matching URL category—Enhanced_Facebook_Apps • Policy action—associate with routing instance RI1	Name of the APBR profile rule. The APBR profile rule matches the traffic to the defined URL categories and redirects the matching traffic to the specified routing instance (example: RI1) for the route lookup.
Category	Enhanced_Social_Web_Facebook	Category defined in the APBR profile rule for matching the traffic.

Table 16: Configuration Parameters for URL Category-Based Routing Using EWF *(Continued)*

Parameters	Name	Description
Routing instance	• Instance name—RI1 • Instance type—forwarding • Static route—1.0.0.254/8 • Next-hop—1.0.0.1	Routing instance of type forwarding is used for forwarding the traffic. All the qualified traffic destined for the static route (with IP address 1.0.0.254/8) is forwarded to the next-hop device (with IP address 1.0.0.1).
RIB group	apbr_group	Name of the RIB group. The RIB group shares interface routes with the forwarding routing instances. To ensure that the next hop is resolvable, interface routes from the main routing table are shared through a RIB group with the routing tables specified in the routing instances.

To perform URL category-based routing using EWF, you must complete the following procedures:

Enabling Enhanced Web Filtering

Step-by-Step Procedure

To use URL categories as match criteria in an APBR profile, you must enable EWF in Content Security.

> **NOTE**: The EWF option requires you to purchase a Juniper Networks Web filtering license. No license is required for local Web filtering.

1. Enable EWF by specifying the Web filtering type as `juniper-enhanced`.

```
[edit]
user@host# set security utm feature-profile web-filtering type juniper-enhanced
```

2. Set the cache size as 500 and cache timeout as 1800 seconds for the configured EWF engine.

   ```
   [edit]
   user@host# set security utm feature-profile web-filtering juniper-enhanced cache size 500
   user@host# set security utm feature-profile web-filtering juniper-enhanced cache timeout 1800
   ```

   For more information about EWF configuration, see Enhanced Web Filtering (EWF).

Defining the Routing Instance and the RIB Group

Step-by-Step Procedure

Define routing instance and the RIB group.

1. Create the routing instance to forward traffic to the different next hops. In this step, you configure the static route 1.0.0.254/8, and the next-hop address as 1.0.0.1.

   ```
   [edit]
   user@host#  set routing-instances RI1 instance-type forwarding
   user@host#  set routing-instances RI1 routing-options static route 1.0.0.254/8 next-hop
   1.0.0.1
   ```

2. Create a RIB group.

   ```
   [edit]
   user@host# set routing-options interface-routes rib-group inet apbr_group
   user@host# set routing-options rib-groups apbr_group import-rib inet.0
   user@host# set routing-options rib-groups apbr_group import-rib RI1.inet.0
   ```

   Interface routes from the main routing table (inet.0) are shared through a RIB group with the routing table specified in the routing instance RI1.inet.0.

Configuring the APBR Profile

Step-by-Step Procedure

Create a rule for the Facebook applications and forward the matching traffic to the routing instance RI1.

1. Create the APBR profile and define the match criteria for the URL category.

   ```
   [edit]
   user@host# set security advance-policy-based-routing profile apbr-pr1 rule rule-social-nw
   match category Enhanced_Social_Web_Facebook
   ```

   The APBR profile rule matches the traffic to the defined URL category—that is, the Facebook application in this example.

2. Specify the action for the traffic matching the URL category.

   ```
   [edit]
   user@host# set security advance-policy-based-routing profile apbr-pr1 rule rule-social-nw then
   routing-instance RI1
   ```

   In this step, you are specifying that the traffic that matches the apbr-pr1 rule is to be redirected to the routing instance RI1.

Configuring the APBR Policy and Attaching the APBR Profile

Step-by-Step Procedure

Associate the application profile to the APBR policy to enable URL category-based routing.

1. Define the APBR policy. Specify the policy match condition as `any` for the source address, destination address, and application.

   ```
   [edit]
   user@host# set security advance-policy-based-routing from-zone trust policy p1 match source-
   address any
   user@host# set security advance-policy-based-routing from-zone trust policy p1 match
   destination-address any
   user@host# set security advance-policy-based-routing from-zone trust policy p1 match
   application any
   ```

   When traffic arrives, it is matched by the APBR policy rules.

2. Attach the APBR profile to the policy.

```
[edit]
user@host# set security advance-policy-based-routing from-zone trust policy p1 then
application-servicesadvance-policy-based-routing-profile apbr-pr1
```

When the traffic matches the APBR policy (p1) rules, the APBR profile apbr-pr1 is applied to the traffic as the action of the APBR policy. The traffic that matches the Facebook application is redirected to the routing instance RI1 according to the APBR profile rule rule-social-nw.

Results

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

`[edit security]`

```
user@host# show advance-policy-based-routing
profile apbr-pr1 {
    rule rule-social-nw {
        match {
            category Enhanced_Social_Web_Facebook;
        }
        then {
            routing-instance RI1;
        }
    }
}
from-zone trust {
    policy p1 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            application-services {
                advance-policy-based-routing-profile apbr-pr1;
            }
        }
    }
}
```

[edit]

```
user@host# routing-options
interface-routes {
    rib-group inet apbr_group;
}
rib-groups {
    apbr_group {
        import-rib [ inet.0 RI1.inet.0 ];
    }
}
```

[edit]

```
user@host# show routing-instances
RRI1 {
    instance-type forwarding;
    routing-options {
        static {
            route 1.0.0.254/8 next-hop 1.0.0.1;
        }
    }
}
```

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

Configuring URL-Based Routing by Using Local Web Filtering

IN THIS SECTION

This section provides the steps to configure URL category-based routing by using local Web filtering.

Table 17 on page 258 provides the details of the parameters used in this example.

Table 17: APBR Configuration Parameters for URL Category-Based Routing Using Local Web Filtering

Parameters	Name	Description
APBR profile	apbr-pr2	Name of the APBR profile.
APBR policy	p2	Name of the APBR policy.
Rule	• Rule name—rule2 • Matching URL category—custom • Policy action—associate with routing instance RI2	Name of the APBR profile rule. The APBR profile rule matches the traffic to the defined URL categories and redirects the matching traffic to the specified routing instance (example: RI2) for the route lookup.
Custom Category (URL Pattern)	203.0.113.0 203.0.113.10	Category defined in the APBR profile rule for matching the traffic.
Routing instance	• Instance name—RI2 • Instance type—forwarding • Static route—5.0.0.10 • Next-hop—9.0.0.1	Routing instance of type forwarding is used for forwarding the traffic. All the qualified traffic destined for the static route (with IP address 5.0.0.10) is forwarded to the next-hop device (with IP address 9.0.0.1).
RIB group	apbr_group2	Name of the RIB group. The RIB group shares interface routes with the forwarding routing instances. To ensure that the next hop is resolvable, interface routes from the main routing table are shared through a RIB group with the routing tables specified in the routing instances.

To perform URL category-based routing using local Web filtering, you must complete the following procedures:

Enabling Local Web Filtering

Step-by-Step Procedure

To use URL categories as match criteria in an APBR profile, you must enable local Web filtering in Content Security.

1. Enable local Web filtering by specifying the Web filtering type as `juniper-local`.

   ```
   [edit]
   user@host# set security utm feature-profile web-filtering type juniper-local
   ```

2. Create custom objects and URL pattern lists.

   ```
   [edit]
   user@host# set security utm custom-objects url-pattern local1 value 203.0.113.0
   user@host# set security utm custom-objects url-pattern local1 value 203.0.113.10
   ```

   In this step, a pattern that matches the IP address 203.0.113.0 or 203.0.113.10 on HTTP is created.

3. Configure the custom URL category list.

   ```
   user@host# set security utm custom-objects custom-url-category custom value local1
   ```

   The URL category specified in this example is custom, where you can add URL lists. In this step, you are adding the URL list `local1`, which includes the patterns matching the addresses 203.0.113.1 and 203.0.113.10 that are created in step "2" on page 259.

4. Configure a Web filtering profile.

   ```
   user@host# set security utm feature-profile web-filtering juniper-local profile P1 category
   custom action permit
   ```

   A Web filtering profile includes a user-defined category with a permit action.

   For more information about local Web filtering configuration, see Local Web Filtering.

Defining the Routing Instance and the RIB Group

Step-by-Step Procedure

Define the routing instance and the RIB group.

1. Create the routing instance to forward traffic to the different next hops. In this example, you configure the static route 5.0.0.0/10, using the next-hop address of 9.0.0.1.

   ```
   [edit]
   user@host#  set routing-instances RI2 instance-type forwarding
   user@host#  set routing-instances RI2 routing-options static route 5.0.0.0/16 next-hop
   9.0.0.1
   ```

2. Create a RIB group.

   ```
   [edit]
   user@host# set routing-options interface-routes rib-group inet apbr_group2
   user@host# set routing-options rib-groups apbr_group2 import-rib inet.0
   user@host# set routing-options rib-groups apbr_group2 import-rib RI2.inet.0
   ```

   Interface routes from the main routing table (inet.0) are shared through a RIB group with the routing table specified in the routing instance (RI2.inet.0).

Configuring the APBR Profile

Step-by-Step Procedure

Create a rule to forward the traffic matching the custom URL pattern to the routing instance RI2.

1. Create the APBR profile and define the match criteria for the URL category.

   ```
   [edit]
   user@host# set security advance-policy-based-routing profile apbr-pr2 rule rule2 match
   category custom
   ```

   The APBR profile rule matches the traffic to the defined custom URL category—that is, traffic with URL patterns matching the addresses 203.0.113.1 and 203.0.113.10 in this example.

2. Specify the action for the traffic matching the URL category.

```
[edit]
user@host# set security advance-policy-based-routing profile apbr-pr2 rule rule2 then routing-
instance RI2
```

In this step, you are specifying that the traffic that matches the rule is to be redirected to the routing instance RI2.

Configuring APBR Policy and Attaching the APBR Profile

Step-by-Step Procedure

Associate the APBR profile to the APBR policy to enable URL category-based routing.

1. Define the APBR policy. Specify the policy match condition as `any` for the source address, destination address, and application.

```
[edit]
user@host# set security advance-policy-based-routing from-zone trust policy p2 match source-
address any
user@host# set security advance-policy-based-routing from-zone trust policy p2 match
destination-address any
user@host# set security advance-policy-based-routing from-zone trust policy p2 match
application any
```

When traffic arrives, is matched by the APBR policy rules.

2. Attach the APBR profile to the policy.

```
[edit]
user@host# set security advance-policy-based-routing from-zone trust policy p2 then
application-services advance-policy-based-routing-profile apbr-pr2
```

When the traffic matches the APBR policy (p2) rules, the APBR profile apbr-pr2 is applied to the traffic as the action of the APBR policy. The traffic that matches the Facebook application is redirected to the routing instance RI2 according to the APBR profile rule rule2.

Results

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

```
[edit security]
```

```
user@host# show advance-policy-based-routing
profile apbr-pr2 {
    rule rule2 {
        match {
            category custom;
        }
        then {
            routing-instance RI2;
        }
    }
}
from-zone trust {
    policy p2 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            application-services {
                advance-policy-based-routing-profile apbr-pr2;
            }
        }
    }
}
```

```
[edit]
```

```
user@host# show routing-options
interface-routes {
    rib-group inet apbr_group2;
}
rib-groups {
    apbr_group2 {
        import-rib [ inet.0 RI2.inet.0 ];
    }
}
```

```
[edit]
```

```
user@host# show routing-instances
RI2 {
    instance-type forwarding;
    routing-options {
        static {
            route 5.0.0.0/10 next-hop 9.0.0.1;
        }
    }
}
```

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

Verification

IN THIS SECTION

Verifying APBR Statistics

Purpose

Display the statistics for APBR, such as the number of sessions processed for the application-based routing, the number of times the APBR is applied for the session, and so on.

Action

From configuration mode, enter the `show security advance-policy-based-routing statistics` command.

`user@host>` **show security advance-policy-based-routing statistics**

```
Advance Profile Based Routing statistics:
  Session Processed:                     5529
```

```
    ASC Success:                           3113
    Rule match success:                    107
    Route  modified:                       107
    AppID Requested:                       2416
```

Meaning

The command output displays the following details:

- Sessions processed for the application-based routing
- The number of times the presence of an entry in the application system cache (ASC) is found
- The number of times the application traffic matches the APBR profile and APBR is applied for the session
- The number of times application identification (AppID) was consulted to identify application traffic
- The number of times the APBR is applied for the session

Bypassing Application Services in an APBR Rule

You can create an APBR profile to include multiple rules with either dynamic applications, application groups or both, or a URL category as match criteria on security devices. URL category-based routing enables you to identify and selectively route Web traffic (HTTP and HTTPS) to a specified destination or to another device where further inspection on the Web traffic is required. In such cases, you can select not to apply or bypass application services on the session that is to be forwarded to the device for further inspection.

Starting in Junos OS Release 19.1R1, you can bypass application services for a session that is re-routed using the APBR rule.

The following sequences are involved in bypassing the application services:

1. APBR uses the application details to look for a matching rule in the APBR profile (application profile).
2. If a matching APBR rule is found, the traffic is redirected to the specified routing instance for the route lookup.
3. If you configure the option to bypass application services on the sessions in an APBR rule, then an attempt is done to bypass the application services to the session.
4. A log message is generated or updated to indicate the bypassing of the application services on the session.

You can bypass the application services including security policies, application quality of service (AppQoS), Juniper ATP Cloud, IDP, Security Intelligence (SecIntel) and Content Security using the APBR rule.

For bypass to be effective, it is required that the APBR rule is matched in the first packet. If the rule is matched after the first packet, and the rule has a bypass option configured, the bypass option is ignored and the application services are not bypassed.

ALG Service is not bypassed due to this feature as bypassing the ALG could potentially result in the correlated (data) session not being matched to appropriate security policy.

Example: Bypassing Application Services by Using APBR Rule

IN THIS SECTION

This example shows you how to bypass application services on the session using APBR rule. Using URL category-based routing, you can identify and selectively route Web traffic (HTTP and HTTPS) to a specified destination or to another device. Here, you can configure to bypass the application services on the session where further inspection on the Web traffic could be performed.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall with Junos OS Release 19.1R1 or later. This configuration example is tested on Junos OS Release 19.1R1.
- Valid application identification feature license installed on the SRX Series Firewall.

Before you begin:

- Define routing instance and RIB group.

- Appropriate security policies to enforce rules for the transit traffic, to specify what traffic can pass through the device, and the actions that need to take place on the traffic as it passes through the device.

Overview

This example shows how to configure APBR on your SRX Series Firewall to forward social media traffic arriving at the trust zone to a specific device or to an interface using URL category-based routing and bypass the application services on the same session.

In this example, you complete the following configurations:

- Define the APBR profile and associate it to a APBR policy. The APBR profile includes the rules to match the traffic with applications and URL categories.
- Next, specify the action of the APBR profile rule. That is, to redirect the matching traffic to the specified routing instance for the route lookup.
- Specify application bypass option for the matching traffic.

When traffic arrives, it is matched by the APBR profile, and if a matching rule is found, the packets are forwarded to the static route. All traffic destined for the static route is transmitted to the next-hop address for transit to a specific device or to an interface. Since you configured application bypass option for the matching traffic, the traffic forwarded to the specific device at next-hop address is not applied with application services.

Configuration

IN THIS SECTION

This section provides steps to configure URL category-based routing by using enhanced Web filtering (EWF) and also enable by passing application services on the traffic.

Enabling Enhanced Web Filtering

Step-by-Step Procedure

To use URL categories as match criteria in an APBR profile, you must enable EWF in Content Security.

> **NOTE**: The EWF option requires you to purchase a Juniper Networks Web filtering license. No license is required for local Web filtering.

1. Enable EWF by specifying the Web filtering type as `juniper-enhanced`.

```
[edit]
user@host# set security utm feature-profile web-filtering type juniper-enhanced
```

2. Set the cache size as 500 and cache timeout as 1800 seconds for the configured EWF engine.

```
[edit]
user@host# set security utm feature-profile web-filtering juniper-enhanced cache size 500
user@host# set security utm feature-profile web-filtering juniper-enhanced cache timeout 1800
```

For more information about EWF configuration, see Enhanced Web Filtering (EWF).

Configuring the APBR Rule

Step-by-Step Procedure

Create a rule for the Facebook applications and forward the matching traffic to the routing instance RI1.

1. Create the APBR profile and define the match criteria for the URL category.

```
[edit]
user@host# set security advance-policy-based-routing profile apbr-pr1 rule rule-social-nw
match category Enhanced_Social_Web_Facebook
```

The APBR profile rule matches the traffic to the defined URL category—that is, the Facebook application in this example.

2. Specify the action for the traffic matching the URL category.

```
[edit]
user@host# set security advance-policy-based-routing profile apbr-pr1 rule rule-social-nw then routing-instance RI1
```

   In this step, you are specifying that the traffic that matches the apbr-pr1 rule is to be redirected to the routing instance RI1.

3. Specify the bypassing application services for the traffic matching the APBR rule.

```
[edit]
user@host# set security advance-policy-based-routing profile apbr-pr1 rule rule-social-nw then application-services-bypass
```

   In this step, you are specifying that the traffic that matches the apbr-pr1 rule is to be bypassed application services.

Configuring APBR Policy and Attaching the APBR Profile

Step-by-Step Procedure

Associate the application profile to the APBR policy to enable URL category-based routing.

1. Define the APBR policy. Specify the policy match condition as `any` for the source address, destination address, and application.

```
[edit]
user@host# set security advance-policy-based-routing from-zone trust policy p1 match source-address any
user@host# set security advance-policy-based-routing from-zone trust policy p1 match destination-address any
user@host# set security advance-policy-based-routing from-zone trust policy p1 match application any
```

   When traffic arrives, it is matched by the APBR policy rules.

2. Attach the APBR profile to the policy.

   ```
   [edit]
   user@host# set security advance-policy-based-routing from-zone trust policy p1 then
   application-services advance-policy-based-routing-profile apbr-pr1
   ```

   When the traffic matches the APBR policy (p1) rules, the APBR profile apbr-pr1 is applied to the traffic as the action of the APBR policy. The traffic that matches the Facebook application is redirected to the routing instance RI1 according to the APBR profile rule rule-social-nw. Also application services are bypassed for the session as specified in APBR profile rule rule-social-nw.

Results

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

```
[edit security]
```

```
user@host# show advance-policy-based-routing
profile apbr-pr1 {
    rule rule-social-nw {
        match {
            category Enhanced_Social_Web_Facebook;
        }
        then {
            routing-instance RI1;
            application-services-bypass;
        }
    }
}
from-zone trust {
    policy p1 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            application-services {
                advance-policy-based-routing-profile apbr-pr1;
            }
        }
```

```
    }
}
```

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

Verification

IN THIS SECTION

Verifying APBR Statistics

Purpose

Display the statistics for APBR, such as the number of sessions processed for the application-based routing, the number of times the APBR is applied for the session, and so on.

Action

From configuration mode, enter the `show security advance-policy-based-routing statistics` command.

`user@host>` **show security advance-policy-based-routing statistics**

```
Advance Profile Based Routing statistics:
  Sessions Processed                      110
  AppID cache hits                        110
  AppID requested                         0
  Rule matches                            2
  Route changed on cache hits             1
  Route changed midstream                 1
  Zone mismatch                           0
  Drop on zone mismatch                   0
  Next hop not found                      0
  Application Services Bypass             1
```

Meaning

The command output displays the following details:

- Sessions processed for the application-based routing
- The number of times the presence of an entry in the application system cache (ASC) is found
- The number of times the application traffic matches the APBR profile and APBR is applied for the session
- The number of times application identification (AppID) was consulted to identify application traffic
- The number of times the APBR is applied for the session
- The number of times the application services are bypassed for the session

Support for User Source Identity in APBR Policies

IN THIS SECTION

Starting in Junos OS Release 19.1R1, you can configure advanced policy-based routing (APBR) policies by defining user source identity as one of the match criteria along with source addresses, destination addresses, and applications. After a successful match, the APBR profile configured with the APBR policy is applied as an application service for the session. The source identity enables you to leverage user information stored in a repository such as user identification table (UIT).

The source-identity field specifies the users and roles to which the policy applies. When the source-identity field is specified in a policy as a matching criterion, user and role information must be retrieved before policy lookup can proceed. Using the source-identity option as a matching criterion in the APBR policy is optional. If the value in the source-identity field is configured as any or there is no entry in the source-identity field, user information and role information are not required and the other match criteria are used for policy lookup.

You can specify one or more users or user roles using the source-identity field with the following keywords:

- authenticated-user—Users that have been authenticated.

- unauthenticated-user—Users that have not been authenticated.
- any—All users regardless of authentication status. If the source-identity field is not configured or is set to any, only other matching criteria are used for matching
- unknown-user—Users that can not be authenticated due to an authentication server disconnection, such as a power outage.

On your security device, the user identification table (UIT) provides user and role information for an active user who has already been authenticated. Each entry in the table maps an IP address to an authenticated user and any role.

UIT contains the IP address, username, and role information for all authenticated user. The entries in the user identification table are ordered by IP address.

On your security device, the type of UIT supported is local authentication table. The local authentication table serves as the authentication source for the information required by APBR policies. Local authentication table is a static UIT created on the device either manually or programmatically using CLI commands. All users included in the local authentication table are considered authenticated users. To retrieve user and role information, a search is performed in the authentication table for an entry with an IP address corresponding to the traffic. When a matching IP address is found, user and role information is retrieved from the table entry and are associated with the traffic. If not found, the user is classified as an unauthenticated user.

User and role information can be created on the device manually or ported from a third-party authentication server, but the data in the local authentication table is not updated in real time.

During APBR policy lookup, if a user and user role that are configured in the APBR policy, but the entry is not present in the local authentication table, then the policy does not match. Hit count value that display the utility rate of security policies according to the number of hits they receive, does not increment.

For more information on user role retrieval and the policy lookup process, see User Role Firewall Security Policies.

Benefits

- Enables you to define the routing behavior at more granular levels to ensure safe enforcement of policy on the application traffic traversing the network.
- Provides more flexible traffic-handling capabilities and offers granular control for forwarding packets based on the roles and business requirements of users.

Local Authentication Table

You can manage the local authentication table with CLI commands that add or delete entries. You can add IP addresses, usernames, and roles from a third-party authentication source to the local authentication table programmatically using CLI commands. If an authentication source defines users and groups, the groups can be configured as roles and associated with the user as usual.

Use the following command to add an entry to a local authentication table. The entries in the table are entered using the IP address.

```
user@host >request security user-identification local-authentication-table add user user-name ip-address ip-address role [role-name role-name ]
```

Example:

```
user@host >request security user-identification local-authentication-table add user-name user1 ip-address 2.2.2.2 roles role1
```

Use the following command to delete an entry by IP address or by username.

```
user@host >request security user-identification local-authentication-table delete (ip-address | user-name)
```

Use the following command to clear the local authentication table:

```
user@host >clear security user-identification local-authentication-table
```

Use the following command to display the content of the local authentication table:

```
user@host >show security user-identification local-authentication-table all (brief | extensive)
```

For more information, see Local Authentication Table.

Example: Configuring Advanced Policy-Based Routing Policies with Source Identity

IN THIS SECTION

This example shows how to configure an APBR policy with source identity and how to apply the APBR profile on a session that matches the APBR policy rules.

Requirements

This example uses the following hardware and software components:

- An SRX Series Firewall with Junos OS Release 19.1R1 or later. This configuration example is tested on Junos OS Release 19.1R1.
- Valid application identification feature license installed on an SRX Series Firewall.

Overview

In this example, you want to forward HTTP traffic arriving at the trust zone to a specific device or interface as specified by the next-hop IP address.

When traffic arrives at the trust zone, it is matched by the APBR policy. When the traffic matches the policy, the configured APBR rule is applied on the permitted traffic as application services. The packets are forwarded based on the APBR rule to the static route and next hop as specified in the routing instance. The static route configured in the routing table is inserted into the forwarding table when the next-hop address is reachable. All traffic destined for the static route is transmitted to the next-hop address for transit to a specific device or interface.

In this example, you must complete the following configurations:

- Define a routing instance and a RIB group.
- Create an ABPR profile.
- Create an APBR policy and attach the APBR profile to it.

Configuration

IN THIS SECTION

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 routing-instances R1 instance-type forwarding
set routing-instances R1 routing-options static route 5.0.0.0/24 next-hop 3.0.0.2
set routing-options interface-routes rib-group inet fbf-group
set routing-options rib-groups fbf-group import-rib inet.0
set routing-options rib-groups fbf-group import-rib RI1.inet.0
set security advance-policy-based-routing profile profile1 rule rule-app1 match dynamic-
application junos:HTTP
set security advance-policy-based-routing profile profile1 rule rule-app1 then routing-instance
R1
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 ge-0/0/1.0
set security advance-policy-based-routing from-zone trust policy SLA1 match source-address any
set security advance-policy-based-routing from-zone trust policy SLA1 match destination-address
any
set security advance-policy-based-routing from-zone trust policy SLA1 match application any
set security advance-policy-based-routing from-zone trust policy SLA1 match source-identity
identity-1
set security advance-policy-based-routing  from-zone trust policy SLA1 then application-services
advance-policy-based-routing-profile profile1
```

Configuring Advanced Policy-Based Routing

Step-by-Step Procedure

To add an entry to a local authentication table.

1. Enter the username, IP address, and user role details.

```
user@host> request security user-identification local-authentication-table add user-name user1
ip-address 2.2.2.2 roles role1
```

Step-by-Step Procedure

To apply APBR on traffic that matches the APBR policy:

1. Create routing instances.

```
[edit]
user@host# set routing-instances R1 instance-type forwarding
user@host# set routing-instances R1 routing-options static route 5.0.0.0/24 next-hop 3.0.0.2
```

2. Group one or more routing tables to form a RIB group called apbr_group and import routes into the routing tables.

```
[edit]
user@host# set routing-options interface-routes rib-group inet fbf-group
user@host# set routing-options rib-groups fbf-group import-rib inet.0
user@host# set routing-options rib-groups fbf-group import-rib RI1.inet.0
```

3. Create the APBR profile and define the rules.

```
[edit]
user@host# set security advance-policy-based-routing profile profile1 rule rule-app1 match
dynamic-application junos:HTTP
user@host# set security advance-policy-based-routing profile profile1 rule rule-app1 then
routing-instance R1
```

4. Create a security zone.

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

5. Create an APBR policy and apply the APBR profile to the security zone.

```
[edit]
user@host# set security advance-policy-based-routing from-zone trust policy SLA1 match source-
address any
user@host# set security advance-policy-based-routing from-zone trust policy SLA1 match
destination-address any
user@host# set security advance-policy-based-routing  from-zone trust policy SLA1 match
application any
user@host# set security advance-policy-based-routing from-zone trust policy SLA1 match source-
identity identity-1
user@host# set security advance-policy-based-routing  from-zone trust policy SLA1 then
application-services advance-policy-based-routing-profile profile1
```

Results

From configuration mode, confirm your configuration by entering the `show routing-instances` 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 routing-instances
R1 {
    instance-type forwarding;
    routing-options {
        static {
            route 5.0.0.0/24 next-hop 3.0.0.2;
        }
    }
}
```

```
[edit]
user@host# show routing-options
interface-routes {
    rib-group inet fbf_group;
}
rib-groups {
    fbf_group {
        import-rib [ inet.0 RI1.inet.0];
    }
}
```

```
[edit]
user@host# show security advance-policy-based-routing
from-zone trust {
    policy SLA1 {
        match {
            source-address any;
            destination-address any;
            application any;
            source-identity identity-1;
        }
        then {
            application-services {
                advanced-policy-based-routing-profile profile1;
            }
        }
    }
}
profile profile1 {
    rule rule-app1 {
        match {
            dynamic-application junos:HTTP;
        }
        then {
            routing-instance R1;
        }
    }
}
```

```
[edit]
user@host# show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
    }
}
```

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

Verification

IN THIS SECTION

Verifying APBR Policy Configuration

Purpose

Display information about the APBR policy, associated APBR profile and to display information about the APBR policy hit count.

Action

From configuration mode, enter the `show security advance-policy-based-routing detail` command.

```
user@host> show security advance-policy-based-routing detail

Policy: SLA1, State: enabled, Index: 5
  Policy Type: Configured
  Sequence number: 1
  From zone: trust
  Source addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Destination addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
      APBR-Profile: profile1
Source identities:
      identity-1
```

Meaning

The command output displays the source identity details in the `Source identities` field.

SEE ALSO

> Understanding Advanced Policy-Based Routing

Using DSCP as Match Criteria in APBR Rules

IN THIS SECTION

This topic includes the following sections:

Introduction

Application identification techniques rely on deep packet inspection (DPI). There are some cases where DPI engine might not be able to identify the application, for example—encrypted traffic. If you apply APBR rules on such traffic, the traffic undergoes normal processing without APBR functionality applied on it.

Starting in Junos OS release 19.3R1, SRX Series Firewalls support configuring DSCP values in an APBR rule as match criteria to perform APBR functionality on the DSCP-tagged traffic.

You can configure DSCP value in addition to the other matching criteria of the APBR rule such as dynamic application, and dynamic application group.

By configuring the DSCP value in an APBR rule, you can extend the APBR service to the traffic with the DSCP markings.

Use Case

You can use APBR rules with DSCP as match criteria for the encrypted traffic.

Limitation

- Support not available for configuring rules with DSCP value and URL category in a single APBR profile.

APBR Rule Lookup When Using a DSCP Value as Match Criteria

In a APBR rule, you can configure a DSCP value or dynamic applications or combination of both.

If you have configured both DSCP and dynamic application in a APBR rule, the rule is considered as match if the traffic matches all the criteria specified in the rule. If there are multiple DSCP values present in the APBR rule, then if any one criteria matches, it is considered as match.

A APBR profile can contain multiple rules, each rule with a variety of match conditions.

In case of multiple APBR rules in a APBR profile, the rule lookup uses the following priority order:

1. Rule with DSCP + dynamic application

2. Rule with dynamic application

3. Rule with DSCP value

If a APBR profile contains multiple rules, the system performs rule lookup and applies the rule in the following order:

- System applies the DSCP-based rules for the first packet of the session.

- System continues to check if any application information available either from DPI classification or application system cache (ASC).

- In the middle of the session, if DPI identifies a new application, the system performs a rule lookup and applies new rule (application-based rule or DSCP-based rule or combination of both) as applicable.

- Identifying application and rule lookup continues till the DPI identifies an application as the final application or maximum reroute value is reached.

- If the rule lookup does not match any rule, no further action is taken.

Lets understand how APBR performs rule lookup and applies the rules with the following two examples:

Example 1

In this example, you configure three APBR rules with— one with DSCP value 30, next rule with application as HTTP, and the third rule with both DSCP value as 30 and application as HTTP. Configure maximum route change value as 1 (default value).

Table 18 on page 283 shows how APBR performs rule lookup and applies the rules.

Table 18: APBR Rules with DSCP and Dynamic Application

Session	Traffic Type	ASC Cache	DPI Classification	Matching Rule
First session	DSCP=30	NA	NA	Rule 1

Midstream session	DSCP=30
	Application = HTTP
		Yes	HTTP	Rule 3
				The traffic switches because rule lookup matched the new rule.
				When traffic switches based on rule change in the middle of the session, the count for maximum route change reduces to 0. Now no further route change takes place in this scenario.

Example 2

In this example, you configure three APBR rules with— one with DSCP value 30, next rule with DSCP value 60, and the third rule with both DSCP value as 30 and application as HTTP.

Table 19 on page 283 shows how APBR performs rule lookup and applies the rules.

Table 19: APBR Rules with Only DSCP Values

Session	Traffic Type	ASC Cache	DPI Classification	Matching Rule
First session	DSCP=30	NA	NA	Rule 1

Midstream session	DSCP=60
	Application = HTTP
		Yes	DSCP=60
			HTTP
				Rule 2
				Rule 3 does not match with traffic because DSCP value is changed from 30 to 60 in midstream.

Configure APBR Rules with DSCP Values as Match Criteria

IN THIS SECTION

This example shows how to configure APBR rules with DSCP values as match criteria.

Configuration

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 zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.1/24
set interfaces ge-0/0/2 unit 0 family inet address 192.0.3.1/24
set interfaces ge-0/0/3 unit 0 family inet address 192.0.4.1/24
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/0/2.0
set security zones security-zone untrust interfaces ge-0/0/3.0
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 ge-0/0/0.0
set interfaces ge-0/0/0 unit 0 family inet address 4.0.0.1/24
set routing-instances RI1 instance-type forwarding
set routing-instances RI1 routing-options static route 192.0.0.0/16 next-hop 192.0.2.254
set routing-instances RI2 instance-type forwarding
set routing-instances RI2 routing-options static route 192.0.0.0/16 next-hop 192.0.3.254
set routing-instances RI3 instance-type forwarding
set routing-instances RI3 routing-options static route 192.0.0.0/16 next-hop 192.0.4.254
set routing-options rib-groups apbr-group import-rib inet.0
set routing-options rib-groups apbr-group import-rib RI1.inet.0
set routing-options rib-groups apbr-group import-rib RI2.inet.0
set routing-options rib-groups apbr-group import-rib RI3.inet.0
set routing-options interface-routes rib-group inet apbr-group
set security advance-policy-based-routing profile p1 rule R1 match dynamic-application junos:HTTP
set security advance-policy-based-routing profile p1 rule R1 then routing-instance RI1
set security advance-policy-based-routing profile p1 rule R2 match dscp 56
set security advance-policy-based-routing profile p1 rule R2 match dynamic-application junos:HTTP
set security advance-policy-based-routing profile p1 rule R2 then routing-instance RI2
set security advance-policy-based-routing profile p1 rule R3 match dscp 46
set security advance-policy-based-routing profile p1 rule R3 then routing-instance RI3
set security zones security-zone trust advance-policy-based-routing-profile p1
set security zones security-zone trust advance-policy-based-routing-profile p1
```

Step-by-Step Procedure

Configure APBR rule with DSCP and dynamic application as match criteria.

1. Define security zones and interfaces.

   ```
   [edit]
   user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.1/24
   user@host# set interfaces ge-0/0/2 unit 0 family inet address 192.0.3.1/24
   user@host# set interfaces ge-0/0/3 unit 0 family inet address 192.0.4.1/24
   user@host# set security zones security-zone untrust host-inbound-traffic system-services all
   user@host# set security zones security-zone untrust host-inbound-traffic protocols all
   user@host# set security zones security-zone untrust interfaces ge-0/0/1.0
   user@host# set security zones security-zone untrust interfaces ge-0/0/2.0
   user@host# set security zones security-zone untrust interfaces ge-0/0/3.0
   ```

2. Define interface and security zones for the ingress interface connecting the client device.

   ```
   [edit]
   user@host# set security zones security-zone trust host-inbound-traffic system-services all
   user@host# set security zones security-zone trust host-inbound-traffic protocols all
   user@host# set security zones security-zone trust interfaces ge-0/0/0.0
   user@host# set interfaces ge-0/0/0 unit 0 family inet address 4.0.0.1/24
   ```

3. Configure the routing instances.

```
[edit]
user@host# set routing-instances RI1 instance-type forwarding
user@host# set routing-instances RI1 routing-options static route 192.0.0.0/16 next-hop 192.0.2.254
user@host# set routing-instances RI2 instance-type forwarding
user@host# set routing-instances RI2 routing-options static route 192.0.0.0/16 next-hop 192.0.3.254
user@host# set routing-instances RI3 instance-type forwarding
user@host# set routing-instances RI3 routing-options static route 192.0.0.0/16 next-hop 192.0.4.254
```

4. Group one or more routing tables to form a RIB group called apbr-group and import routes into the routing tables.

```
[edit]
user@host# set routing-options rib-groups apbr-group import-rib inet.0
user@host# set routing-options rib-groups apbr-group import-rib RI1.inet.0
user@host# set routing-options rib-groups apbr-group import-rib RI2.inet.0
user@host# set routing-options rib-groups apbr-group import-rib RI3.inet.0
user@host# set routing-options interface-routes rib-group inet apbr-group
```

5. Define the APBR rule with dynamic application HTTP as match criteria.

```
[edit]
user@host# set security advance-policy-based-routing profile p1 rule R1 match dynamic-application junos:HTTP
user@host# set security advance-policy-based-routing profile p1 rule R1 then routing-instance RI1
```

   APBR routes the traffic matching the HTTP application to the routing instance RI1.

6. Create another rule for DSCP and HTTP application.

```
[edit]
user@host# set security advance-policy-based-routing profile p1 rule R2 match dscp 56
user@host# set security advance-policy-based-routing profile p1 rule R2 match dynamic-application junos:HTTP
```

```
user@host# set security advance-policy-based-routing profile p1 rule R2 then routing-instance
RI2
```

APBR routes the traffic matching the DSCP value 56 to the routing instance RI2.

7. Define one more rule with DSCP value 46.

```
[edit]
user@host# set security advance-policy-based-routing profile p1 rule R3 match dscp 46
user@host# set security advance-policy-based-routing profile p1 rule R3 then routing-instance
RI3
user@host# set security zones security-zone trust advance-policy-based-routing-profile p1
```

APBR routes the traffic matching the DSCP value 46 to the routing instance RI3.

8. Apply the APBR profile to the security zone.

```
[edit]
user@host# set security zones security-zone trust advance-policy-based-routing-profile p1
```

Results

From configuration mode, confirm your configuration by entering the `show security advance-policy-based-routing`, `show routing-instances`, 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 interfaces
ge-0/0/0 {
    unit 0 {
        family inet {
            address 4.0.0.1/24;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 192.0.3.1/24;
        }
    }
}
ge-0/0/3 {
    unit 0 {
        family inet {
            address 192.0.4.1/24;
        }
    }
}
```

```
[edit]
user@host# show routing-instances
RI1 {
    instance-type forwarding;
    routing-options {
        static {
            route 192.0.0.0/16 next-hop 192.0.2.254;
        }
    }
}
RI2 {
    instance-type forwarding;
    routing-options {
        static {
            route 192.0.0.0/16 next-hop 192.0.3.254;
        }
    }
}
RI3 {
    instance-type forwarding;
    routing-options {
        static {
            route 192.0.0.0/16 next-hop 192.0.4.254;
        }
    }
}
```

```
[edit]
user@host# show security advance-policy-based-routing
profile p1 {
    rule R1 {
        match {
            dynamic-application junos:HTTP;
        }
        then {
            routing-instance RI1;
        }
    }
    rule R2 {
        match {
            dynamic-application junos:HTTP;
            dscp 56;
        }
        then {
            routing-instance RI2;
        }
    }
    rule R3 {
        match {
            dscp 46;
        }
        then {
            routing-instance RI3;
        }
    }
}
```

```
[edit]
user@host# show security zones
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/1.0;
        ge-0/0/2.0;
        ge-0/0/3.0;
    }
}
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        ge-0/0/0.0;
    }
    advance-policy-based-routing-profile {
        p1;
    }
}
```

Once you complete the configuration, enter `commit` from configuration mode.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall with Junos OS Release 19.3R1 or later. This configuration example is tested on Junos OS Release 19.3R1.
- Any supported SRX Series Firewall.
- Valid application identification feature license installed on the SRX Series Firewall.

Overview

In this example, you want to forward HTTP traffic and traffic tagged with DSCP value 56 and DSCP value 46 to a specific device or interfaces at Site 1, Site 2, and Site 3 respectively. Security device forwards the traffic based on an application or DSCP value to a preferred route by using APBR feature.

When traffic arrives at the trust zone, APBR matches the traffic with configured APBR profile rules. If the traffic matches the rule, APBR forwards the traffic to the specific destination as defined in the APBR rule.

For example, you configure APBR to route the traffic to different destinations based on the type of the application as specified below:

- Rule 1—Forward HTTP traffic from Client 1 to the Site 1 using next-hop address 192.0.2.254.
- Rule 2—Forward traffic with DSCP value 56 and HTTP application to Site 2 using next-hop device 192.0.3.254.
- Rule 3—Forward traffic with DSCP value 46 to Site 3 using the next-hop device 192.0.4.254.

Figure 11 on page 291 shows the topology used in this example.

Figure 11: Topology for Advanced Policy-Based Routing (APBR) Configuration

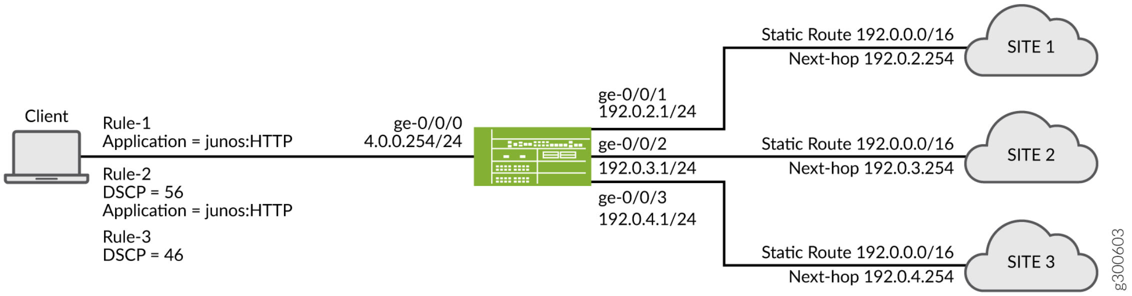


Table 20 on page 291 provides the details of the parameters used in this example.

Table 20: Configuration Parameters

Parameter	Value	Associated Parameter	Description
APBR profile	P1	Name of the APBR profile.	Configure the profile with rules to match the applications and DSCP values and specify destination (example: routing-instances) for the matching traffic.
RIB group	RI1.inet.0	Associated routing instance—RI1	Configure the RIB group to import interface route entries from inet.0, RI1.inet.0, RI2.inet.0, and RI3.inet.0.

Table 20: Configuration Parameters *(Continued)*

Parameter	Value	Associated Parameter	Description
	RI1.inet.2	Associated routing instance—RI2	
	RI1.inet.3	Associated routing instance—RI3	
Routing instance	RI1	• Static route—192.0.0.0/16 • Next-hop—192.0.2.254	Configure the routing instances to include next-hop IP address. APBR forwards the qualified traffic destined for the static route to the next-hop device address in Site 1, Site 2, and Site 3.
	RI2	• Static route—192.0.0.0/16 • Next-hop—192.0.3.254	
	RI3	• Static route—192.0.0.0/16 • Next-hop—192.0.4.254	
APBR Rule	R1	• Matching application—junos:HTTP • Associated routing instance—RI1	Configure the APBR rules and specify dynamic application or DSCP values as matching criteria. APBR forwards the matching traffic to the associated routing instance.
	R2	• matching DSCP value— 56 and application—junos:HTTP. • Associated routing instance—RI2	

Table 20: Configuration Parameters *(Continued)*

Parameter	Value	Associated Parameter	Description
	R3	• matching DSCP value— 46 • Associated routing instance—RI3	

Verification

IN THIS SECTION

Verifying Advanced Policy-Based Routing Statistics

Purpose

Display the statistics for APBR such as the number of sessions processed for the application-based routing, number of times the APBR is applied for the session, and so on.

Action

From configuration mode, enter the `show security advance-policy-based-routing statistics` command.

```
user@host> show security advance-policy-based-routing statistics
```

```
Advance Profile Based Routing statistics:
  Sessions Processed                    0
  App rule hit on cache hit             0
  App rule hit on HTTP Proxy/ALG        0
  URL cat rule hit on cache hit         0
  DSCP rule hit on first packet         0
  App and DSCP hit on first packet      0
```

```
    App rule hit midstream                 0
    URL cat rule hit midstream             0
    App and DSCP rule hit midstream        0
    DSCP rule hit midstream                0
    Route changed on cache hits            0
    Route changed on HTTP Proxy/ALG        0
    Route changed midstream                0
    Zone mismatch                          0
    Drop on zone mismatch                  0
    Next hop not found                     0
    Application services bypass            0
```

Meaning

The command output displays the following details:

- Sessions processed for the application-based routing.
- The number of times the application traffic or DSCP-tagged traffic matches the APBR profile.
- The number of times traffic is switched to different route in the midstream.

Verifying Advanced Policy-Based Routing Sessions

Purpose

Display information about the sessions and packet flows active on the device, including detailed information about specific sessions.

Action

From configuration mode, enter the `show security flow session` command to display information about all currently active security sessions on the device.

Meaning

The command output displays the following details:

- All active sessions and packet flows on your device.
- List of incoming and outgoing IP flows, including services.

- Security attributes associated with a flow, for example, the policies that apply to traffic belonging to that flow.

- Session timeout value, when the session became active, how long the session has been active, and if there is active traffic on the session.

Disable APBR Midstream Routing for Specific APBR Rule

IN THIS SECTION

Why Selectively Disabling the Midstream Routing is Required?

Some sessions go through continuous classification in the middle of the session as application signatures identify the application. Whenever an application is identified by the application signatures, APBR is applied, and this results in a change in the route of the traffic. You can limit the number of times a route can change for a session by using the `max-route-change` option. If you set this option to 0, the APBR is disabled for the particular session. However, this option also disables the APBR functionality globally on your device which might not be required.

Selectively Disabling APBR In Midstream

Starting in Junos OS Release 19.4R1, you can selectively turn-off the APBR service in the middle of a session for a specific APBR rule, while retaining the global APBR functionality for the remaining sessions. When you disable midstream routing for a specific APBR rule, the system does not apply midstream APBR for corresponding application traffic, and routes the traffic through a non-APBR route.

To selectively disable the midstream APBR, you can configure the APBR rule with disable midstream routing option (`disable-midstream-routing`) at [**edit security advance-policy-based-routing profile *apbr-profile-name* rule *apbr-rule-name***] hierarchy level.

Table 21 on page 296 shows the behavior of the selectively disabling midstream APBR option.

Table 21: Selectively Disabling APBR in Midstream for Different Scenarios

Traffic Type	Traffic Matches APBR Rule	Result
New Sessions (when the cache entry does not exists for the session)	With `disable-midstream-routing` option	Session uses the default route.
		The `max-route-change` value is not decremented.
	Without `disable-midstream-routing` option	Apply midstream APBR
		Apply APBR till the last application is identified or as defined in the `max-route-change` option.
Established Sessions (when the cache entry exists for the session)	With `disable-midstream-routing` option	Apply APBR.
		Disengage APBR for the further sessions. That is—even if further applications are identified in the session after the cache hit, APBR is not applied to them.
	Without `disable-midstream-routing` option	Apply APBR.
		Continue to apply APBR till the last application is identified or as defined in the `max-route-change` option.

Disabling midstream routing for a specific APBR rule will reroute the application traffic back through a default non-APBR route.

Using Disable Midstream Routing Option to Selectively Disable APBR for Specific APBR Rule

If you have already configured an APBR rule for a specific application, and now you want to selectively disable the APBR midstream routing, use the following option:

```
user@host# set security advance-policy-based-routing profile apbr-profile-name rule apbr-rule-name disable-midstream-routing
```

Example:

```
[edit security advance-policy-based-routing]
    user@host# show
    profile p1 {
    rule r1 {
        disable-midstream-routing;
        match {
            dynamic-application junos:YAHOO;
        }
        then {
            routing-instance RI1;
        }
    }
}
from-zone trust {
    policy policy-1 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            application-services {
                advance-policy-based-routing-profile profile-1;
            }
        }
    }
}
```

Use the `show security advance-policy-based-routing statistics` command to verify the APBR status:

```
Advance Profile Based Routing statistics:
  Sessions Processed                        9
  App rule hit on cache hit                 0
  App rule hit on HTTP Proxy/ALG            0
  Midstream disabled rule hit on cache hit  2
  URL cat rule hit on cache hit             0
  DSCP rule hit on first packet             2
  App and DSCP hit on first packet          0
  App rule hit midstream                    1
  Default rule match                        0
  Midstream disabled rule hit midstream     1
  URL cat rule hit midstream                0
  App and DSCP rule hit midstream           0
  DSCP rule hit midstream                   0
  Route changed on cache hits               2
  Route changed on HTTP Proxy/ALG           0
  Route changed midstream                   0
  Default rule applied                      0
  Zone mismatch                             0
  Drop on zone mismatch                     0
  Next hop not found                        0
  Application services bypass               0
```

In this sample output, the fields `Midstream disabled rule hit on cache hit` and `Midstream disabled rule hit midstream` indicate the number of times a route remains unchanged in the middle of a session after the rule with defined application is matched and the number of times the rule with a disabled midstream has a matching entry in the application system cache (ASC).

Default Mechanism to Forward the Traffic Through APBR Rule

Starting in Junos OS 20.1R1 Release, you can configure “any” as match criteria for dynamic application in a APBR rule. The criteria “any” acts as a wildcard and applies to any dynamic application.

Example

```
user@hots# set security advance-policy-based-routing profile p1 rule R1 match dynamic-application
any
user@hots# set security advance-policy-based-routing profile p1 rule R1 then routing-instance RI1
```

Application traffic that match the other parameters in a APBR rule matches the policy regardless of the dynamic application type.

Note the following while using the any keyword for dynamic applications in an APBR rule:

- You can configure only one APBR rule with any keyword for the dynamic application in an APBR profile.
- Configuring a same APBR rule with DSCP and URL-based categories with the any keyword is not supported.
- APBR rule with dynamic applications configured as any is applied only during the first packet processing.
- Configuring a same APBR rule with dynamic application as any and other dynamic applications or dynamic application groups is not supported.

Release History Table

Release	Description
21.3R1	Support for first packet inspection is available starting in Junos OS Release 21.3R1 and later releases.
19.4R1	Starting in Junos OS Release 19.4R1, you can selectively turn-off the APBR service in the middle of a session for a specific APBR rule, while retaining the global APBR functionality for the remaining sessions
19.3R1	Starting in Junos OS release 19.3R1, SRX Series Firewalls support configuring DSCP values in an APBR rule as match criteria to perform APBR functionality on the DSCP-tagged traffic
19.1R1	Starting in Junos OS Release 19.1R1, you can bypass application services for a session that is re-routed using the APBR rule.
19.1R1	Starting in Junos OS Release 19.1R1, you can configure advanced policy-based routing (APBR) policies by defining user source identity as one of the match criteria along with source addresses, destination addresses, and applications

17.4	Starting with Junos OS Release 15.1X49-D110 and Junos OS Release 17.4R1, SRX Series Firewalls support advanced policy-based routing (APBR) with an additional enhancement to apply the APBR in the middle of a session (which is also known as midstream support)
15.1X49-D60	Starting with Junos OS Release 15.1X49-D60, SRX Series Firewalls support advanced policy-based routing (APBR)
15.1X49-D123	Support for reverse rerouting is available starting in Junos OS Release 15.1X49-D130 and later releases.

RELATED DOCUMENTATION

Application Identification | 5
Application Firewall | 130
Application Tracking | 166
Application QoS | 190

Application Quality of Experience

IN THIS SECTION

Application Quality of Experience (AppQoE)

IN THIS SECTION

Introduction to Application Quality of Experience

The relentless growth of cloud computing, mobility, and Web-based applications, requires that the network identify and control the traffic at the application level, and handle each application type separately to provide quality of experience (QoE) for users. To ensure application-specific QoE (AppQoE), you need to effectively prioritize, segregate, and route application traffic without compromising performance or availability.

AppQoE utilizes (or employs) the capabilities of two application security services - application identification (AppID) and advanced policy-based routing (APBR). It uses AppID to identify specific applications in your network and advanced policy-based routing (APBR) to specify a path for certain traffic by associating SLA profiles to a routing instance on which the application traffic is sent as per APBR rules.

One of the important requirements of a software-defined WAN (SD-WAN) is to measure the quality of underlay network paths and, based on the results, determine the best paths to use for the delivery of each packet. AppQoE monitors the performance of business- critical applications, and based on the score, selects the best possible link for that application traffic in order to meet performance requirements specified as in SLA (service-level agreement).

The presence of an SLA rule in the APBR configuration triggers the AppQoE functionality; If there are no SLA profiles available, the APBR functions without triggering AppQoE.

Supported Use Case

You can configure AppQoE optimally using Contrail Service Orchestration (CSO). We recommend that you use CSO to configure AppQoE for Juniper Networks Contrail SD-WAN solution. For more details, see Application Quality of Experience Overview and Configure and Monitor Application Quality of Experience.

Supported SRX Series Firewalls

AppQoE is supported on both hub-and-spoke and full mesh topologies in SD-WAN deployments.

You can configure vSRX Virtual Firewall instances, SRX300 line devices, SRX550M as spoke devices and SRX1500, SRX4100 and SRX4200 as hub devices.

You can configure an AppQoE between two SRX Series Firewall endpoints (book-ended) and both SRX Series Firewalls must have the same version of the Junos OS image.

Starting in Junos OS Release 15.1X49-D160 and in Junos OS 19.1R1, SRX4100 and SRX4200 devices support AppQoE when the devices are in chassis cluster mode. You can configure the device to operate both in active/active and in active/passive modes and deploy the device as spoke device in SD-WAN deployments.

Benefits of Application Quality of Experience

- Enables cost-effective QoE by providing real-time monitoring of application traffic to provide a consistent and predictable level of service.

- Increases customer retention and satisfaction by providing a guaranteed SLA for the delivery of the certain traffic (such as video traffic). AppQoE ensures that the approved traffic receives the appropriate priority, and bandwidth required to ensure the best quality of experience to the user.

Limitations

Implementation of AppQoE on security devices has the following limitations:

- All the different routes to the destination through different interfaces must have the same preference, weight, and metrics configured. All routes must be added as ECMP paths for the destination and must also be part of the same forwarding table.

- AppQoE SLA service only between two security devices endpoints (book-ended) are supported. End-to-end AppQoE SLA service is not supported.

- AppQoE can be applied only if all interfaces are part of the same zone.

- AppQoE cannot be applied for reverse traffic.

- AppQoE does not influence in change in the destination for a session.

- AppQoE does not support IPv6/UDP probe encapsulation, GRES, chassis cluster (ISSU, high-availability, dual CPE high availability, Z-mode high availability), and logical systems.

- AppQoE does not support preferred path selection and transit virtual routing and forwarding (VRF) are not supported.
- AppQoE does not support passive probing on IPv6 data packets.
- An input firewall filter is required at the non-WAN interfaces to discard UDP packets with UDP destination port 36000.
- The SRX4600 device has the following limitations:
  - The class of service (CoS) for generic routing encapsulation (GRE) is not supported when AppQoE is configured.
  - Passive probe details might not be available for the each short-lived session.
  - Synchronization of the session states might not happen on secondary node in Z-line mode traffic processing when device is operating in chassis cluster mode.

How Application Quality of Experience Works?

AppQoE utilizes AppID and APBR capabilities to identify specific applications/application groups and specify a path for certain traffic by associating SLA profiles to a routing instance on which the application traffic is sent as per APBR rules.

AppQoE monitors the performance of applications, and based on the score, selects the best possible link for that application traffic in order to meet performance requirements specified as in SLA (service-level agreement).

Identifying Applications or Application Groups

Following steps are involved in identifying applications or application groups:

1. Junos OS application identification identifies applications and once an application is identified, its information is saved in the application system cache (ASC).
2. APBR evaluates the packets based to determine if the session is candidate for application-based routing (advance policy-based routing). If this is first packet of the new session and traffic is not flagged for application-based routing, it undergoes normal processing (non-APBR route) to destination.
3. If the session needs application-based routing, APBR queries the ASC module to get the application attributes (IP address, destination port, protocol type, and service).
4. If the application in ASC is found, traffic is further processed for a matching rule in the APBR profile.
   - If a matching rule is found, the traffic is redirected to the specified routing instance for the route lookup.

  - AppQoE checks whether an SLA is enabled for a session. If the session is a candidate for an SLA measurement, AppQoE initiates active and passive probes for performance measurements.
  - If SLA is not enabled for the session in the APBR rule, the AppQoE ignores that session and the default behavior of APBR is applied to those sessions—that is, traffic is routed through the specified routing instance for the destination.
- If the application in is not found in ASC, APBR requests for deep inspection of the flow. that is, application signature package is installed and application identification for the session is enabled, so that ASC can be populated for use by subsequent sessions for APBR processing (see step 2).

Specifying Path for Applications or Application Groups

The following steps summarize how AppQoE specifies a path for the application traffic according to the SLA rules.

1. APBR uses the application details to look for a matching rule in the APBR profile (application profile). Traffic matching the applications and application groups, are forwarded to the static route and the next-hop address as specified in the routing instance.
2. An SLA rule attached to the APBR profile specifies parameters, that are required to measure the SLA and to identify whether any SLA violation has occurred or not.
3. The applications traffic is assigned to a particular overlay link based on the SLA metrics of that overlay link measured using active probing.
4. The SLA violation is determined through passive probing of live application/application group traffic. The best path/overlay link for the application/application group is determined through the path selection algorithm.

Application Traffic Path Selection

The following steps take place for routing data traffic from source to destination, specifically, to select the best path,

- For the first data packet of a flow (first path), if the application is already known (from the ASC lookup), then the best path for the application is searched in the database. If the application is not known or is new (from ASC lookup), then a random path or the default path is chosen. This path continues for the entire session. Later, after the application is detected by the DPI, the database is updated with the best path for the application.
- For the remaining data packet of a flow (fast path), if the application is not known initially, then the particular session continues on the same path. If the application is known initially, then AppQoE selects the best path for the application traffic.

When a new application is detected, the path selection mechanism attempts to find a path that satisfies all the SLA metrics. If no such path exists, then the next best path (based on number of metrics satisfied) is used. If there are more than one path that satisfies the metrics, a random path among the available paths is selected. The SLA violation is detected when any one of the metric is violated or none of the metrics meets the requirement, based on the profile configuration.

How Application Quality of Experience Measures Application Performance

Application performance is determined by the following indicators:

- Latency—The amount of time physically required for media to travel depending on media length and distance that need to be covered
- RTT— A round-trip time required to travel from source to destination and vice versa.
- Packet loss—Packet loss reflects the number of packets lost per 100 of packets sent by a host.
- Jitter—Jitter is the difference in the latency from packet to packet. Ingress jitter, egress jitter, and two-way jitter can be specified for evaluating the performance of the link.

AppQoE monitors RTT, jitter, and packet loss on each link, and based on the score, seamlessly diverts applications to the alternate path if performance of the primary link is below acceptable levels as specified by SLA. Measurement and monitoring of application performance is done using active and passive probes to detect SLA violations and to select an alternate path for that particular application.

AppQoE collects real-time data by continuously monitoring application traffic and identifying network or device issues by:

- Monitoring the performance on all configured overlay links.
- Using passive probes (inline with the application datapath) and active probes (synthetic probes for specific application) to monitor the traffic performance for application or application group.
- Sending all collected performance metrics or metadata for analysis to a log collector.
- Comparing specified application against a specific performance metric and changing the path for the application traffic dynamically in case of an SLA violation.
- Supporting flexible SLA metric configuration for a given application or application group.

AppQoE measures the application SLA across multiple WAN links, and maps the application traffic to a path among the available links, that is, to the path that best serves the SLA requirement.

Application Performance Measurement by Using Active and Passive Probes

Active and passive probe measurements are the two approaches used for end-to-end analysis of the network.

- Active probe—Active probes measure the service quality of the application to provide an end-to-end measurement of the network performance.

  In active probing, custom packets are sent between spoke and hub points on all the multiple routes and the RTT, latency, jitter, and packet-loss are measured between the installed probe points. The active probes are sent periodically on all the active and passive links. A configured number of samples is collected and a running average for each such application's probe path is measured. If there is a violation detected for any application traffic, the probe metrics are evaluated to determine the best link that satisfies the SLA.

- Passive probe—Passive probes are installed on links within the network, and they monitor all the traffic that flows through those links.

  Passive probing monitors links for SLA violations on live data traffic. In a passive probe, the actual data packets are encapsulated in an IP/UDP probe header in the live traffic between the SRX Series book-ended points, and RTT, jitter and packet loss between the points of installation of the probes are measured to compute the service quality.

  If there is a violation detected for any application, the synthetic probe metrics are evaluated to determine the best link that satisfies the SLA.

  NOTE: Starting in Junos OS Release 18.3R1 and in Junos OS Release 15.1X49-D150, on all supported SRX Series Firewalls and vSRX Virtual Firewall instances, in order to detect if a link or path is down by passive probes, a minimum of three probe requests and 100% packet loss must occur in a sampling period for a given session to trigger SLA violation.

  NOTE: When the device is operating in chassis cluster mode, if the secondary node (node 1), through which traffic is forwarded, is rebooted, multiple switching of the application traffic between the links across secondary node links occurs. This happens when the available links on primary node(node 0) are having less active probe SLA path score compared to the secondary node links. This behavior continues until AppQoE active probe SLA path score results are available to indicate that there is 100% packet loss on all the links on secondary node.

You can configure an SLA rule with active and passive probe parameters and associate the SLA rule with APBR profile. The APBR profile also includes a APBR rule. Rules are associated with one or more than one application or application groups and the traffic matching the rule is redirected to the routing instance

AppQoE triggers the probe requests to all probe paths of the application. Active and passive probes monitor the network for areas or points of failures or congestion.

AppQoE collects traffic class statistics for learned applications using active and passive probes and takes following actions:

1. Measure performance for SLA—The real-time metrics provided by probes are used to score service quality according to the SLA for an application and determine whether the application path does not meet SLA requirements. That is, if there is a violation detected for any application, the synthetic probe metrics are evaluated to determine the best alternate link for the application traffic that satisfies the SLA.

2. Reroute traffic—Switch the application traffic between the two links, that is, when one link has performance issues, the traffic is routed to the other link during the same session.

NOTE: If the application's traffic can be reachable through multiple links, you must configure all the reachable paths as overlay paths and attach the overlay paths to application's SLA rule.

Switching Application Traffic to An Alternate Path

You can enable or disable switching of the application traffic to another route (local to the device) during an SLA violation. When local route switching is enabled, switching of the application traffic to an alternate route is enabled and the SLA monitoring and reporting functionality is also available. Even when the option for switching of the application traffic to an alternate path is disabled in the SLA rule configuration, AppQoE resolves SLA violations---for example, by switching the application traffic to a new path

When local route switching is disabled, only SLA monitoring and reporting functionality is available and switching of the application traffic to the different route because of an SLA violation is tuned off.

When an application traffic switches to an alternative path, there will be a short time period during which the application traffic cannot be switched again to another path in case of SLA violation. This time period helps to avoid flapping of the traffic across links.

Understanding AppQoE Configuration Limits

Starting in Junos OS Release 15.1X49-D160 and in Junos OS Release 19.1R1, AppQoE enforces the configuration limit for overlay paths, metric profiles, probe parameters, and SLA rules per profile when you configure application-specific SLA rules and associates the SLA rules to an APBR profile.

If you configure the parameters more than the allowed limit, error messages are displayed when you commit the configuration.

Examples of error messages:

The following sample error messages are from the SRX4100 and SRX4200 device. The value of the configuration limit might not reflect exact number supported; the numbers might differ between the supported devices

```
[edit security advance-policy-based-routing]
  'sla-rule sla0'
    Cannot configure more than 32 sla rules
error: configuration check-out failed
```

```
[edit security advance-policy-based-routing]
  'overlay-path grep2'
    Cannot configure more than 2000 overlay paths
error: configuration check-out failed
```

```
[edit security advance-policy-based-routing]
  'metrics-profile m0'
    Max metrics for this system is 32
error: configuration check-out failed
```

```
[edit security advance-policy-based-routing]
  'active-probe-params pr0'
    Cannot configure more than 64 probe params
error: configuration check-out failed
```

Application Path Selection Based on Link Preference and Priority

IN THIS SECTION

One of the important requirements of a software-defined WAN (SD-WAN) is to measure the quality of underlay network paths and, based on the results, determine the best paths to use for the delivery of each packet.

Starting in Junos OS Release 18.4R1 and in Junos OS Release 15.1X49-D160, you can configure application-specific quality of experience (AppQoE) to select the application path based on the link priority and the link type when multiple paths that meet the SLA requirements are available.

You can select an MPLS or Internet link as the preferred path, assign the priority between 1 through 255 with a lower value indicating a more preferred link. A value of one (1) indicates highest priority. If there are multiple paths available, the path which has the highest priority is selected.

For example, If an MPLS path is selected for VoIP traffic and quality degradation occurs during a call because of jitter or packet loss, the packets are sent through another path (Internet) that meets SLA requirements. Now application traffic is sent through the Internet path and if the quality in the Internet path is degraded, the path is switched back to MPLS.

You can configure the link priority and link type of each underlay interface in an advanced policy-based routing (APBR) rule, and the same parameters are inherited by the corresponding overlay. An underlay interface in this case is the final outgoing interface in the routing topology for the overlay.

For example, in a network infrastructure, if the underlay is a fourth-generation (4G) LTE connection, then the dialer interface can be configured as the underlay interface for AppQoE. Similarly, if the underlay is a DSL connection, then the corresponding Point-to-Point Protocol over Ethernet (PPPoE) interface can be configured as the underlay interface for AppQoE.

Starting in Junos OS Release 21.2R1, the AppQoE path selection mechanism is enhanced with custom link tag configuration, application traffic switch to the higher priority link of the preferred tags, non-SLA metrics based deployment, and overlay interface attribute preference features.

Benefits of Application Path Preference and Priority

- Provides flexibility of selecting the best path for for application traffic.

- Enables routing of application traffic over the cost-effective connectivity option while ensuring SLA requirements (latency and jitter) are met.
- Supports dynamic path switching if the selected application path experiences a degradation in quality.

Path Selection Mechanism

Application traffic is routed through separate links based on the link preference as following:

- AppQoE path selection mechanism includes a list of best paths to a specific destination that meets the SLA requirements. From this list, AppQoE selects a path that matches the link preference configured by the user.
- If there are multiple such paths, the path that has the highest priority among them is selected.
- If there is no priority or link type preference configured, then a random path or the default path is selected.
- If no links that meet the SLA requirements are available, then the best available link in terms of the highest SLA score and link type preference, in case strict affinity is configured, is selected.
- If multiple links that meet the SLA requirements are available, then the one with the highest priority is selected.

System Log Messages for AppQoE

IN THIS SECTION

Starting in Junos OS Release 19.2R1, the support for the application-level logging is available for AppQoE on SRX Series Firewalls. This feature is introduced to reduce the impact on CSO or log collector device while processing large number of system log messages generated at the session-level. The security device maintains session-level information and provides system log messages for the session level. With application-level logging replacing session-level logging, the overhead on security device decreases and AppQoE log throughput increases.

AppQoE sends following system log messages:

- APPQOE_SLA_METRIC_VIOLATION: When a violation is detected for a session and when a session's path is resolved as a result of moving to a new link.
- APPQOE_BEST_PATH_SELECTED: When a session switches the path for its data traffic.

With application-level logging, all session-level logs are supported at the application-level. The AppQoE functionality of sending real-time probes, measuring the SLA metrics, violation detection, and path-switch continues at the session-level. However, as part of application-level summarization feature, datapath sessions notify the SLA metrics, violation information, and path switch to AppQoE database. The information thus received from datapath is aggregated at the application-level, and then sent in the form of system logs to collector device.

Table 22 on page 311 provides details of new application-level logs are supported from Junos OS Release 19.2R1 onwards.

Table 22: Application-Level Log Messages

system log Message	Description
APPQOE_APP_SLA_METRIC_VIOLATION	• This system log message is generated the first time the application is in violation. • The SLA metrics are measured for each application session in the data path. The SLA violation metrics continue to be measured at the session-level only. However, the metrics or data pertaining to the SLA violation are sent to the AppQoE database by all data sessions of that application when their SLA is violated. • In the case of dual CPE, the node which is active for the application generates the APPQOE_APP_SLA_METRIC_VIOLATION report.
APPQOE_APP_BEST_PATH_SELECTED	• This system log message is generated when an application goes through a path switch. This log report is also generated to clear the violation happened because of self heal (when the SLA violation is cleared by itself before any change in the link) • For application-level logging, Once an application or a link switches to an alternate path, AppQoE sends the log message APPQOE_APP_BEST_PATH_SELECTED to the collector device.

Table 22: Application-Level Log Messages *(Continued)*

system log Message	Description
APPQOE_APP_PASSIVE_SLA_METRIC_REPORT	• This system log message is generated for passive probe SLA metrics data. This message is generated once the number of samples collected meet with the SLA export factor. • With the support of application-level logging, each probe candidate session sends information to AppQoE where the metrics are aggregated and averaged out before it is sent to the collector. Therefore the passive SLA report thus aggregated at the application level includes the averaged data from all of those application data sessions.

Application-level logging introduces the following AppQoE functionality changes:

- Active probe maintains and uses only real-time RTT and jitter values. For packet loss, it refers the previous session’s cause because packet loss can be calculated only at the end of the window.
- During configuration commit, active probe sets RTT and jitter values to highest 32-bit value for all entries.
- Active probe retains previous session’s values until the a proper real-time value of the metrics are available.
- When a 100% packet loss is experienced in active probing, all other metrics are set to highest 32-bit value.

Reporting of Invalid Values for RTT and Jitter

When the data for RTT and Jitter is not available, log messages sent with an invalid value of 0xFFFFFFFF and it can be ignored by the log collector. Table 23 on page 312 provides some possible scenarios when the invalid RTT and Jitter is sent.

Table 23: Application-Level Log Messages Affected by Invalid Data for RTT and Jitter

Scenario	Affected System Logs
100% packet loss:	APPQOE_APP_PASSIVE_SLA_METRIC_REPORT APPQOE_APP_SLA_METRIC_VIOLATION

Table 23: Application-Level Log Messages Affected by Invalid Data for RTT and Jitter *(Continued)*

Scenario	Affected System Logs
Packet-loss greater than 0 and less than 100%:	APPQOE_APP_PASSIVE_SLA_METRIC_REPORT APPQOE_APP_SLA_METRIC_VIOLATION
No Packet-loss	APPQOE_APP_SLA_METRIC_VIOLATION APPQOE_APP_PASSIVE_SLA_METRIC_REPORT

Disable AppQoE Logging

By default AppQoE log-type is set as system log. If you want to disable AppQoE, then configure the log-type as disabled in the following configuration:

1. Disable AppQoE logging

```
[edit]
user@host# set security advance-policy-based-routing sla-options log disabled
```

2. Enable AppQoE logging

```
[edit]
user@host# set security advance-policy-based-routing sla-options log system log
```

Application Quality of Experience (AppQoE) Based on the DSCP Bits of Incoming Traffic

IN THIS SECTION

To overcome this scenario, Starting in Junos OS Release 19.4R1, AppQoE supports SLA-based path selection for the incoming traffic on the basis of DSCP value. AppQoE selects the best possible link for the application traffic based on the application signature or DSCP value or combination of both application identification and DSCP value. See

DSCP Support in APBR

When you configure both DSCP and dynamic application in a APBR rule, the rule is considered as match if the traffic matches all the criteria specified in the rule. When there are multiple DSCP values present in the APBR rule, then if any one criteria matches, it is considered as match.

A APBR profile can contain multiple rules, each rule with a variety of match conditions.

In case of multiple APBR rules in a APBR profile, the rule lookup uses the following priority order:

1. Rule with DSCP + dynamic application

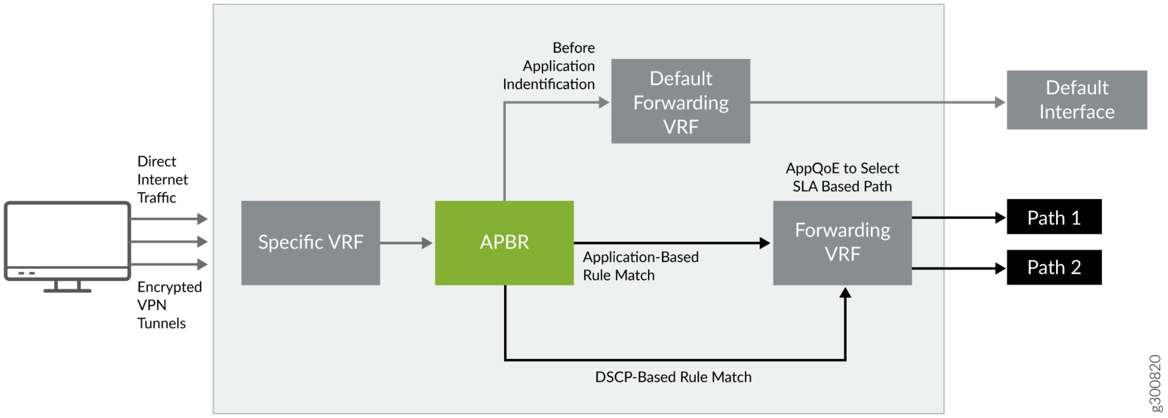
2. Rule with dynamic application

3. Rule with DSCP value

Network Service Orchestrator can map application to DSCP value at external service function and the same is provisioned at the gateway router to map the DSCP to desired SLA profile.

Figure 12 on page 314 shows a scenario where AppQoE performs SLA-based path selection for the incoming traffic on the basis of DSCP value and application signature in a gateway router use case.

Figure 12: Path Selection for the Traffic Based on DSCP Value and Application



For the traffic based on the DSCP value, AppQoE works as follows:

- All the traffic entering the gateway router from LAN undergoes application identification. Until DPI identifies an application, the system forwards the traffic stream to a default forwarding virtual routing and forwarding (VRF) instance. VRF includes an outgoing interface associated to it.
- Junos OS application identification identifies applications and once an application is identified, its information is saved in the application system cache (ASC).
- The system continues to check if any application information available either from DPI classification or ASC.
- The APBR mechanism classifies sessions based on well-known applications signatures and DSCP values and uses policy to identify the best possible route for the application. The APBR policy maps application traffic to a specific VRF.
- The presence of an SLA rule in the APBR configuration triggers the AppQoE functionality; AppQoE performs SLA-based path selection for the traffic based on the application or DSCP value.

A single DSCP includes multiple application categories bundled into it. Different application categories have their individual traffic pattern. In such a scenario, detection of violation using passive probes and applying it to all the sessions might cause false negative and false positive. As a workaround, avoid using passive probing when you have configured DSCP-based SLA rule. You can use active probes for the destination path group to which the traffic is forwarded.

Limitations

AppQoE deployments with DSCP-based rules on the device is chassis cluster mode have the following limitations:

- If the rule match is completed before the application identification is done, and AppQoE moves the session to the other node, then application identification does not complete. This condition occurs when the DSCP-based rule is configured.
- If you have configured two APBR rules—1) with DSCP value 2) with both DSCP and dynamic application, and assigned a same DSCP value in both the rules, on receiving the first packet, APBR matches with the DSCP rule. In case the best path is identified on the other node, then the session is moved to the other node. In this scenario, the application sessions are matched against the DSCP rule and not with the APP+DSCP rule.

APBR Policies for AppQoE

Starting in Junos OS Release 20.1R1, AppQoE utilizes the granular rule matching functionality provided by APBR to provide the quality of experience (QoE) based on the application traffic.

In Junos OS Release 18.2R1, APBR supported configuring policies by defining source addresses, destination addresses, and applications as match conditions. After a successful match, the configured APBR profile is applied as an application services for the session. In Junos OS Release 20.1R1, AppQoE leverages the APBR enhancement and selects the best possible link for the application traffic as sent by APBR to meet performance requirements specified in SLA.

For example, you want to forward Telnet and HTTPS traffic arriving at the trust zone to a specific device or interface through a best available link. When traffic arrives at the trust zone, APBR matches the traffic with matching criteria source address, destination address and applications defined in APBR policies. If traffic matches the policy, corresponding APBR profiles are applied.

APBR uses the application details to look for a matching rule in the profile. If a matching rule is found, the traffic is redirected to the specified routing instance as defined in the rule.

AppQoE Multi-homing with Active-Active Deployment

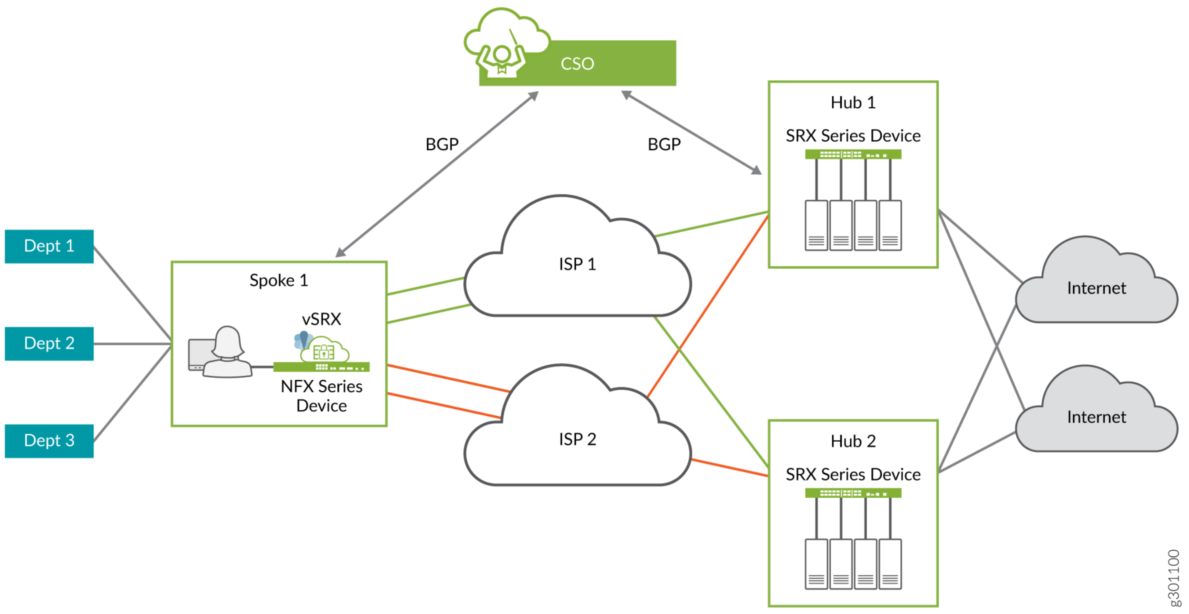
IN THIS SECTION

Starting In Junos OS Release 20.2R1, AppQoE is enhanced to support multi-homing with active-active deployment. Previously, AppQoE supported multihoming with active-standby deployment.

In active-active deployment, the spoke device connects to multiple hub devices. Application traffic can transit through any of the hub devices if the link to the hub device meets SLA requirements. Application traffic can seamlessly switch between the hub devices in case of SLA violation or the active hub device is not responding

Figure 1 shows a mesh topology. In this topology, an end point is reachable through more than one node.

Figure 13: A Sample Mesh Topology



To enable multihoming in active-active mode, you must configure the BGP multipath to allow the device to select multiple equal-cost BGP paths to reach a given destination.

When you enable BGP multipath, the device selects multiple equal-cost BGP paths to reach a given destination, and all these paths are installed in the forwarding table. AppQoE completes the route lookup and gets the next-hop route details along with the corresponding overlay-links. AppQoE obtains the overlay-link property from the configured destination path group.

Based on the application’s SLA requirements and link preferences, AppQoE determines the best link among all the links in that destination-path-group. In case of SLA violation, based on the SLA score and link preferences, AppQoE selects alternate links across all the configured destination-path-group if the end-point is reachable through those links.

For more information on BGP multipath configuration, see Examples: Configuring BGP Multipath.

Limitation

In certain scenario when next-hop ID for the route changes, the existing sessions remain on the SLA-violated link even though another link that meets SLA requirements is available. However, the new sessions are not impacted in this case and they are routed through the links that meet SLA requirements.

Support for SaaS Applications

IN THIS SECTION

Starting in Junos OS Release 20.4R1, we've extended application quality of experience (AppQoE) support for Software as a Service (SaaS) applications.

AppQoE performs service-level agreement (SLA) measurements across the available WAN links such as underlay, GRE, IPsec or MPLS over GRE. It then sends SaaS application data over the most SLA-compliant link to provide a consistent service.

Support for IPv6 Traffic

- Starting in Junos OS Release 21.3R1, you can use IPv6 addresses in AppQoE configurations. The support includes:
  - IPv6 address in overlay path configuration
  - Active probing sessions using IPv6 addresses as source and destination address.
  - IPv4 and IPv6 traffic from the client side
  - Dual stacking of IPv4 and IPv6 on the LAN side
  - IPv6 address on the LAN side for SaaS (software as a service) probing.

    For SaaS probing, ensure that you configure both IPv4 and IPv6 addresses for the incoming interface for IPv4 and IPv6 interoperability.
- Starting in Junos OS Release 21.4R1, you can use dual stacking of IPv4 and IPv6 for overlay and underlay networks in an AppQoE configuration.

Release History Table

Release	Description
19.4R1	Starting in Junos OS Release 19.4R1, AppQoE supports SLA-based path selection for the incoming traffic on the basis of DSCP value

19.2R1	Starting in Junos OS Release 19.2R1, the support for the application-level logging is available for AppQoE on SRX Series Firewalls.
19.1R1	Starting in Junos OS Release 15.1X49-D160 and in Junos OS 19.1R1, AppQoE is supported on SRX4100 and SRX4200 device when the device is operating in chassis cluster mode
19.1R1	Starting in Junos OS Release 15.1X49-D160 and in Junos OS Release 19.1R1, AppQoE enforces the configuration limit for overlay paths, metric profiles, probe parameters, and SLA rules per profile when you configure application-specific SLA rules and associates the SLA rules to an APBR profile.

RELATED DOCUMENTATION

Advanced Policy-Based Routing | 220

Application Identification | 5

Application-Based Multipath Routing

IN THIS SECTION

Application-Based Multipath Routing Overview

IN THIS SECTION

Traffic for video and voice are sensitive to packet loss, latency and jitter. Packet loss directly leads to degradation in the quality of voice and video calls. in voice or video calls.

To ensure timely delivery of these sensitive application traffic, application-based multipath routing (also referred as multipath routing in this document) is supported on SRX Series Firewalls to allow the sending device to create copies of packets, send each copy through two or more WAN links.

Multipath identifies two or more paths based on the SLA configuration and sends out a copy of the original traffic on all the identified paths.

On the other end, among the multiple copies of the packet received, the receiving device selects the first received packet and drops the subsequent ones. On the receiving device, while the copy of the packet is in progress, multipath calculates the jitter and packet loss for the combined links and then estimates the jitter and packet loss for the same traffic on individual links. You can compare the reduction in packet loss when combined links are used instead of individual links used for traffic.

Sending the multiple copies of the application traffic ensures that if there is a packet loss or delay, the other link might still deliver the packet to the endpoint.

SRX Series Firewalls support application-based multipath routing starting in Junos OS Release 15.1X49-D160 in standalone mode.

SRX Series Firewalls support application-based multipath routing starting in Junos OS Release 19.2R1 and Junos OS Release 15.1X49-D170 in chassis cluster mode.

Multipath routing leverages following functionality:

- Application identification details from Deep Packet Inspection(DPI)
- APBR functionality for packet forwarding feature
- AppQoE service for SLA association.

Supported Use Cases

- SD-WAN hub and spoke topology
- SD-WAN mesh topology

Limitations

- All the selected WAN links must be of ECMP paths for a destination.
- All the selected WAN interfaces which need to be a part of multipath routing sessions must belong to one single zone
- Multipath routing feature is supported only between two book-ended security devices.

Benefits of Multipath Routing

- Multipath support in SD-WAN uses case enhances application experience by reducing packet loss, faster delivery of the packet, and less jitter that results in better quality of service for the traffic especially for the voice and video traffic.

Understanding Workflow in Multipath Routing

The following sequences are involved in applying multipath routing:

- Junos OS application identification identifies applications and once an application is identified, its information is saved in the application system cache (ASC).
- Application policy-based routing (APBR) queries the application system cache (ASC) module to get the application attributes details.
- APBR uses the application details to look for a matching rule in the APBR profile (application profile). If a matching rule is found, the traffic is redirected to the specified routing instance for the route lookup.
- AppQoE checks whether an SLA is enabled for a session. If the session is candidate for an SLA measurement, and if multipath routing is configured, then multipath routing is triggered.
- Based on the SLA rule, multipath routing obtains the underlay link types and corresponding overlays on which packet duplication needs to be performed. Multipath routing can be triggered based on the configuration of an SLA rule. When multipath routing is configured within an SLA rule for a specific application, AppQoE functionality is disabled for all sessions of that application matching the SLA rule.
- Based on the application traffic and the configured bandwidth limit, multipath identifies two or more paths and triggers a copy of the original traffic on all the identified paths. Multipath routing path selection is done on the overlay paths. The parameters to limit the bandwidth is based on the underlay link-speed and selection is based on link-type.

- On the receiving device, while the copy of the packet is in progress, multipath calculates the jitter and packet loss for the combined links and then estimates the jitter and packet-loss for same traffic on individual links.
- On the receiving device, multipath routing accepts packets of a session arriving through different links, maintain sequence of a packet arriving on different CoS queues, and drop any duplicates.

Multipath routing copies packets on all the links belonging to a rule till the bandwidth limit is reached. The bandwidth limit is calculated based on the least link speed identified for that rule. This is applicable for all the sessions for all the applications which match that multipath routing rule. Once the limit is reached, multipath routing stops copying of packets and starts a timer for a time period as configured in max-time-wait option in the multipath routing configuration. When the timer expires, it restarts the copying of the packets again.

AMR Improvements

IN THIS SECTION

Starting in Junos OS Release 21.2R1, following enhancements are introduced for AMR:

AMR Support for Reverse Traffic

you can apply multipath functionality on the reverse traffic. Now both the sending device and the receiving device can create copies of packets, and send each copy through two WAN links to the destination device. This enhancement ensures uninterrupted delivery of the sensitive application traffic at both directions.

By default, AMR for the reverse traffic is disabled. You can enable it with the following CLI option:

```
set security advance-policy-based-routing multipath-rule rule-name enable-reverse-wing
```

To disable AMR for the reverse traffic, use the following CLI option:

```
delete security advance-policy-based-routing multipath-rule rule-name enable-reverse-wing
```

AMR support for the reverse wing traffic is available when the devices are operating in HA mode. Note that the packets in the queue are dropped during HA failover.

Queuing Mechanism for Out-of-Order Packets

Starting in Junos OS Release 21.2R1, queuing mechanism for the out-of-order packets at the receiving device is improved.

Previously, the AMR receiving device discarded out-of-order packets resulting in packet loss and degrade in the quality-of-service. With the queuing mechanism, when the receiving device receives out-of-order packets, it further waits for some more packets to arrive, and then buffers those packets in the queue for short duration. This buffering helps in reordering of packets and prevents discarding of packets.

AMR Support for APBR Profile

Starting in Junos OS Release 21.2R1, the security device supports AMR when used with a APBR profile configured with a APBR policy. You can create the APBR policy by defining source addresses, destination addresses, and applications as match conditions.

In the previous releases of Junos OS, you could attach an APBR profile to an incoming security zone of the ingress traffic. In this case, the APBR was applied per security zone basis.

Following example shows configuration snippet of a APBR policy by defining source addresses, destination addresses, and applications as match conditions. An SLA rule is applied for the traffic matching APBR policy rules. A multipath rule associated with the SLA rule gets applied and multipath routing functionality is enabled for the session.

```
set security advance-policy-based-routing multipath-rule amr-rule1 number-of-paths 2
set security advance-policy-based-routing multipath-rule amr-rule1 bandwidth-limit 30
set security advance-policy-based-routing multipath-rule amr-rule1 max-time-to-wait 60
set security advance-policy-based-routing multipath-rule amr-rule1 application junos:SSH
set security advance-policy-based-routing multipath-rule amr-rule1 application junos:HTTP
set security advance-policy-based-routing multipath-rule amr-rule1 link-type MPLS
set security advance-policy-based-routing multipath-rule amr-rule1 link-type IP
set security advance-policy-based-routing profile apbr1 rule rule1 match dynamic-application
junos:RTP
set security advance-policy-based-routing profile apbr1 rule rule1 then routing-instance TC1_VPN
set security advance-policy-based-routing profile apbr1 rule rule1 then sla-rule sla1
set security advance-policy-based-routing sla-rule sla1 multipath-rule amr-rule1
set security zones security-zone trust advance-policy-based-routing-profile apbr1
set security advance-policy-based-routing from-zone trust policy sla_policy1 match source-
address 10.4.0.1
set security advance-policy-based-routing from-zone trust policy sla_policy1 match destination-
address 10.5.0.1
set security advance-policy-based-routing from-zone trust policy sla_policy1 match application
junos-RTP
set security advance-policy-based-routing from-zone trust policy sla_policy1 then application-
services advance-policy-based-routing-profile apbr1
```

Link Selection

In previous releases, for application-based multipath routing, the link selection mechanism was either default (one of the first two available links) or based on the link type (IP/MPLS) configuration AppQoE underlay-interface configuration.

Starting in Junos OS Release 21.2R1, you can specify the link preference options as generic routing encapsulation (GRE) and secure tunnel (st). The device directly selects one of the specified interfaces for multipath routing.

If you have not configured the `link-preference`, then the AMR selects links from the first two available links from the configured paths.

You can specify link preferences using the following CLI option:

```
set security advance-policy-based-routing multipath-rule rule-name link-preferences [st0.0 |
st0.1}
```

AMR in SLA Violation Mode or Standalone Mode

Starting in Junos OS Release 21.2R1, AMR is enabled in one of the following two modes:

- SLA violation mode—When the AppQoE detects SLA violation on all the links, it enables the AMR. AMR is disabled when SLA is met on any of the links based on the timer configuration .

- Standalone mode—When you've configured AMR without configuring SLA metrics, then AMR is enabled independent of AppQoE status. In this mode, when bandwidth limit is reached, then AMR is paused for a default duration and then restarted.

Example:

Following is a samp configuration of an SLA metrics. SLA metrics specifies requirement parameters, which are used by AppQoE to evaluate the SLA of the link. To accomplish the SLA, AppQoE monitors the network for sources of failures or congestion. If the performance of a link is below acceptable levels as specified by the SLA, the situation is considered as an SLA violation. If the LA violation is noticed on all the links, AMR is enabled in SLA violation mode.

```
set security advance-policy-based-routing metrics-profile metric1 sla-threshold delay-round-trip
50000
set security advance-policy-based-routing metrics-profile metric1 sla-threshold jitter 10000
set security advance-policy-based-routing metrics-profile metric1 sla-threshold jitter-type
egress-jitter
set security advance-policy-based-routing metrics-profile metric1 sla-threshold packet-loss 4
set security advance-policy-based-routing metrics-profile metric1 sla-threshold match all
set security advance-policy-based-routing sla-rule sla1 metrics-profile metric1
```

If the SLA metrics configuration (as shown in example above) is not available in the AMR configuration, then AMR is enabled in standalone mode.

Support for IPv6 Traffic

Application-based multipath routing supports IPv6 traffic:

- IPv6 traffic over IPv4 tunnels (Junos OS Release 21.2R1)

- IPv6 traffic over IPv6 tunnels (Junos OS Release 21.3R1)

Support AMR over IPsec and Generic Routing Encapsulation (GRE) Sessions

- Application-based multipath routing over direct IPsec tunnels without GRE (Junos OS Release 21.2R1)

- Application-based multipath routing over direct Generic Routing Encapsulation (GRE) tunnels without IPsec (Junos OS Release 21.2R1)

- Application-based multipath routing over direct IPsec tunnels without GRE for IPv6 traffic (Junos OS Release 21.3R1)

- Application-based multipath routing over direct GRE tunnels without IPsec for IPv6 traffic (Junos OS Release 21.3R1)
- Application-based multipath routing over MPLS-over-GRE-over-IPsec for IPv6 traffic (Junos OS Release 21.3R1)

SEE ALSO

multipath-rule

interface (advance-policy-based-routing)

Application-Based Multipath Routing Sample Configuration

IN THIS SECTION

Sample application based multipath routing configuration (hub and spoke topology)

This section covers sample application based multipath routing configuration for hub and spoke topology. The configuration uses the SLA set by the APBR and works independent of APPQoE. For APPQoE SLA, see Application Quality of Experience . You can configure the device for additional features like link selection based on preference, path selection based on link type, and multipath routing support over IPsec and GRE tunnels. Multipath routing can be configured with Contrail Service Orchestrator. See Contrail Service Orchestration (CSO) Deployment Guide for details.

Spoke side device basic configuration

```
user@host# set security advance-policy-based-routing profile profile1 rule r1 match dynamic-application junos:HTTP
user@host# set security advance-policy-based-routing profile profile1 rule r1 match dynamic-application junos:SIP
user@host# set security advance-policy-based-routing profile profile1 rule r1 then routing-instance TC1_VPN
user@host# set security advance-policy-based-routing profile profile1 rule r1 sla-rule sla_rule1
```

```
user@host# set security advance-policy-based-routing sla-rule sla_rule1 multipath-rule mult1
user@host# set security advance-policy-based-routing multipath-rule mult1 number-of-paths 2
user@host# set security advance-policy-based-routing multipath-rule mult1 bandwidth-limit 90
user@host# set security advance-policy-based-routing multipath-rule mult1 application junos:HTTP
user@host# set security advance-policy-based-routing multipath-rule mult1 application junos:SIP
```

Hub side device basic configuration

```
user@host# set security advance-policy-based-routing multipath-rule mult1 number-of-paths 2
user@host# set security advance-policy-based-routing multipath-rule mult1 bandwidth-limit 90
user@host# set security advance-policy-based-routing multipath-rule mult1 enable-reverse-wing
user@host# set security advance-policy-based-routing multipath-rule mult1 application junos:HTTP
user@host# sset security advance-policy-based-routing multipath-rule mult1 application junos:SIP
```

Link preference configuration

```
user@host# set security advance-policy-based-routing multipath-rule mult1 link-preferences
gr-0/0/0.0
user@host# set security advance-policy-based-routing multipath-rule mult1 link-preferences
gr-0/0/0.1
```

Link type based path selection configuration

```
user@host# set security advance-policy-based-routing multipath-rule mult1 link-type MPLS
user@host# set security advance-policy-based-routing multipath-rule mult1 link-type IP
```

Interface based configuration at application based multipath routing level

```
user@host# set security advance-policy-based-routing interface gr-0/0/0.0 link-tag IP
user@host# set security advance-policy-based-routing interface gr-0/0/0.1 link-tag MPLS
```

IPsec VPN configuration with IPv6 tunnels and IPv4 traffic at spoke side device for application based multipath routing

```
user@host# set groups ipsec-groups security ike proposal salausehdotp1 authentication-method pre-
shared-keys
user@host# set groups ipsec-groups security ike proposal salausehdotp1 dh-group group5
user@host# set groups ipsec-groups security ike proposal salausehdotp1 encryption-algorithm
aes-256-gcm
```

```
user@host# set groups ipsec-groups security ike policy salauspolitiikkap1 mode main
user@host# set groups ipsec-groups security ike policy salauspolitiikkap1 proposals salausehdotp1
user@host# set groups ipsec-groups security ike policy salauspolitiikkap1 pre-shared-key ascii-
text "$9$1-7ESeLxd2oGdbPQnCB1-VwYgJDi.TF/aZ"
user@host# set groups ipsec-groups security ike gateway gateway1 ike-policy salauspolitiikkap1
user@host# set groups ipsec-groups security ike gateway gateway1 version v2-only
user@host# set groups ipsec-groups security ipsec proposal salausehdotp2 protocol esp
user@host# set groups ipsec-groups security ipsec proposal salausehdotp2 encryption-algorithm
aes-256-gcm
user@host# set groups ipsec-groups security ipsec policy salauspolitiikkap2 perfect-forward-
secrecy keys group5
user@host# set groups ipsec-groups security ipsec policy salauspolitiikkap2 proposals
salausehdotp2
user@host# set groups ipsec-groups security ipsec vpn vpn1 df-bit clear
user@host# set groups ipsec-groups security ipsec vpn vpn1 ike ipsec-policy salauspolitiikkap2
user@host# set groups ipsec-groups security ipsec vpn vpn1 establish-tunnels immediately
user@host# set system host-name SRX345-2
user@host# set system root-authentication encrypted-password "$ABC123"
user@host# set system services ssh root-login allow
user@host# set services application-identification
user@host# set security apply-groups ipsec-groups
user@host# set security ike gateway SRX345-1-A address fdf:a::1
user@host# set security ike gateway SRX345-1-A external-interface ge-0/0/0.0
user@host# set security ike gateway SRX345-1-B address fdf:b::1
user@host# set security ike gateway SRX345-1-B external-interface ge-0/0/1.0
user@host# set security ipsec vpn SRX345-1-A bind-interface st0.0
user@host# set security ipsec vpn SRX345-1-A ike gateway SRX345-1-A
user@host# set security ipsec vpn SRX345-1-B bind-interface st0.1
user@host# set security ipsec vpn SRX345-1-B ike gateway SRX345-1-B
user@host# set security application-tracking first-update
user@host# set security application-tracking session-update-interval 1
user@host# set security forwarding-options family inet6 mode flow-based
user@host# set security forwarding-options family mpls mode flow-based
user@host# set security flow allow-dns-reply
user@host# set security flow allow-embedded-icmp
user@host# set security flow sync-icmp-session
user@host# set security flow tcp-mss all-tcp mss 1300
user@host# set security flow tcp-mss ipsec-vpn mss 1350
user@host# set security flow tcp-session no-syn-check
user@host# set security flow tcp-session no-syn-check-in-tunnel
user@host# set security flow tcp-session no-sequence-check
user@host# set security policies default-policy permit-all
user@host# set security zones security-zone Untrust host-inbound-traffic system-services all
```

```
user@host# set security zones security-zone Untrust host-inbound-traffic protocols all
user@host# set security zones security-zone Untrust interfaces ge-0/0/0.0
user@host# set security zones security-zone Untrust interfaces ge-0/0/1.0
user@host# set security zones security-zone Untrust application-tracking
user@host# set security zones security-zone Untrust enable-reverse-reroute
user@host# set security zones security-zone trust host-inbound-traffic system-services all
user@host# set security zones security-zone trust host-inbound-traffic protocols all
user@host# set security zones security-zone trust interfaces ge-0/0/6.0
user@host# set security zones security-zone trust application-tracking
user@host# set security zones security-zone trust advance-policy-based-routing-profile apbr
user@host# set security zones security-zone trust enable-reverse-reroute
user@host# set security zones security-zone VPN host-inbound-traffic system-services all
user@host# set security zones security-zone VPN host-inbound-traffic protocols all
user@host# set security zones security-zone VPN interfaces st0.0
user@host# set security zones security-zone VPN interfaces st0.1
user@host# set security zones security-zone VPN application-tracking
user@host# set security zones security-zone VPN enable-reverse-reroute
user@host# set security zones security-zone test interfaces ge-0/0/7.0 host-inbound-traffic
system-services all
user@host# set security zones security-zone test interfaces ge-0/0/7.0 host-inbound-traffic
protocols all
user@host# set security advance-policy-based-routing profile apbr rule r1 match dynamic-
application junos:ICMP
user@host# set security advance-policy-based-routing profile apbr rule r1 match dynamic-
application junos:SSH
user@host# set security advance-policy-based-routing profile apbr rule r1 match dynamic-
application junos:ICMP-ECHO
user@host# set security advance-policy-based-routing profile apbr rule r1 then routing-instance
apbr
user@host# set security advance-policy-based-routing profile apbr rule r1 then sla-rule sla1
user@host# set security advance-policy-based-routing sla-rule sla1 multipath-rule amr-rule1
user@host# set security advance-policy-based-routing multipath-rule amr-rule1 number-of-paths 2
user@host# set security advance-policy-based-routing multipath-rule amr-rule1 bandwidth-limit 30
user@host# set security advance-policy-based-routing multipath-rule amr-rule1 enable-reverse-wing
user@host# set security advance-policy-based-routing multipath-rule amr-rule1 max-time-to-wait 60
user@host# set security advance-policy-based-routing multipath-rule amr-rule1 application
junos:ICMP
user@host# set security advance-policy-based-routing multipath-rule amr-rule1 application
junos:SSH
user@host# set security advance-policy-based-routing multipath-rule amr-rule1 application
junos:ICMP-ECHO
user@host# set security advance-policy-based-routing multipath-rule amr-rule1 link-preferences
st0.0
```

```
user@host# set security advance-policy-based-routing multipath-rule amr-rule1 link-preferences
st0.1
user@host# set interfaces ge-0/0/0 unit 0 family inet6 address fdf:a::2/64
user@host# set interfaces ge-0/0/1 unit 0 family inet6 address fdf:b::2/640
user@host# set interfaces ge-0/0/6 unit 0 family inet address 10.0.12.1/24
user@host# set interfaces ge-0/0/7 unit 0 family inet address 10.0.12.10/24
user@host# set interfaces fxp0 unit 0 family inet address 192.168.123.2/24
user@host# set interfaces st0 unit 0 family inet address 10.0.0.2/30
user@host# set interfaces st0 unit 1 family inet address 10.0.1.2/30
user@host# set policy-options policy-statement ecmp then load-balance per-packet
user@host# set routing-instances IPSEC protocols ospf area 0.0.0.0 interface st0.0 interface-type
p2p
user@host# set routing-instances IPSEC protocols ospf area 0.0.0.0 interface st0.1 interface-type
p2p
user@host# set routing-instances IPSEC protocols ospf area 0.0.0.0 interface ge-0/0/6.0 passive
user@host# set routing-instances IPSEC interface ge-0/0/6.0
user@host# set routing-instances IPSEC interface st0.0
user@host# set routing-instances IPSEC interface st0.1
user@host# set routing-instances IPSEC instance-type virtual-router
user@host# set routing-instances IPSEC routing-options interface-routes rib-group inet apbr-group
user@host# set routing-instances apbr instance-type forwarding
user@host# set routing-instances apbr routing-options static route 0.0.0.0/0 next-hop st0.0
user@host# set routing-instances apbr routing-options static route 0.0.0.0/0 next-hop st0.1
user@host# set routing-instances test interface ge-0/0/7.0
user@host# set routing-instances test instance-type virtual-router
user@host# set routing-instances test routing-options static route 0.0.0.0/0 next-hop 10.0.12.1
user@host# set routing-options rib-groups apbr-group import-rib IPSEC.inet.0
user@host# set routing-options rib-groups apbr-group import-rib apbr.inet.0
user@host# set routing-options forwarding-table export ecmp
```

NOTE: For GRE tunnels replace `ipsec` with `gre`. For IPv4 tunnel, IPv4 traffic and IPv6 traffic, replace the configuration with IPv4 and IPv6 appropriately.

Example: Configuring Application-Based Multipath Routing

IN THIS SECTION

This example shows how to configure multipath routing to provide quality of experience (QoE) by enabling real-time monitoring of the application traffic according to the specified SLA.

Requirements

- Supported SRX Series Firewall with Junos OS Release 15.1X49-D160, Junos OS Release 19.2R1, or later. This configuration example is tested for Junos OS Release 15.1X49-D160.
- Valid application identification feature license installed on a security device.
- Appropriate security policies to enforce rules for the transit traffic, in terms of what traffic can pass through the device, and the actions that need to take place on the traffic as it passes through the device.
- Enable application tracking support enabled for the zone. See "Application Tracking" on page 166.
- Ensure that following features are configured:
  - Application Identification
  - APBR
  - AppQoE
  - Link Preference and Priority for AppQoE

Overview

To ensure uninterrupted delivery of these sensitive application traffic, application-based multipath routing is supported on security devices to allow the sending device to create copies of packets, and send each copy through two WAN links to the destination.

Multipath routing identifies two paths based on the SLA configuration and creates duplicate copy of the application traffic and sends the traffic simultaneously on different physical paths. On the receiving device, while the copy of the packet is in progress, multipath routing estimates on the reduction in jitter, RTT and packet loss and analyzes the quality of service for routing the traffic to the best link to provide SLA to the end user. This also helps in estimation on the reduction in jitter, RTT and packet loss is done. If both the copies are received on the remote end, then the first received packet is considered, and drops the subsequent ones.

Table 24 on page 332 provides the details of the parameters used in this example.

Table 24: Configuration Parameters for Multipath Rule, SLA Rule, and APBR

Parameter	Options	Values
Multipath rule (multi1)	Number of paths	2
	bandwidth-limit	60
	Maximum time to wait	60
	Link type	MPLS, IP
	application	junos:YAHOO, junos:GOOGLE
	application-group	junos:web
SLA rule (sla1)	Associated multipath rule	multi1
APBR profile (apbr1)	Match applications	junos:YAHOO
	APBR rule	rule1
	SLA rule	sla1
	Underlay interface	ge-0/0/2 and ge-0/0/3 • Speed: 800 Mbps

In this example, you configure a multipath rules for junos:YAHOO and junos:GOOGLE application traffic. Then configure an SLA rule and associate multipath rules with multipath rule.

Next, associate the SLA rules with APBR rules created for the Yahoo application. APBR uses the application details to look for a matching rule in the APBR profile (application profile).

Multipath rule is applied on the traffic matching junos:YAHOO or junos:GOOGLE, and forwarded to and the next-hop address as specified in the routing instance.

Multipath routing obtains the underlay link types and corresponding overlays on which packet duplication is required based on the SLA rule. Based on the application traffic and the configured bandwidth limit, multipath identifies two or more paths and triggers a copy of the original traffic on all the identified paths.

When traffic reaches on receiving end, the receiving device accepts packets of a session arriving through different links, and maintains sequence of a packet arriving on different CoS queues and drops any duplicate packets.

NOTE: Ensure that configuration is the same across the devices on both the sending-side and on the receiving-side device is such that devices can to act as both sender and a receiver.

Configuration

IN THIS SECTION

Configure Multipath Rules for Application Traffic (Device Configured to Send Traffic)

Step-by-Step Procedure

Configure APBR profiles for different applications traffic and associate SLA rule and multipath rule.

1. Create routing instances.

```
user@host# set routing-instances TC1_VPN instance-type vrf
user@host# set routing-instances TC1_VPN route-distinguisher 10.150.0.1:101
user@host# set routing-instances TC1_VPN vrf-target target:100:101
```

```
user@host# set routing-instances TC1_VPN vrf-table-label
user@host# set routing-instances TC1_VPN routing-options static route 10.19.0.0/8 next-table
Default_VPN.inet.0
```

2. Group one or more routing tables to form a RIB group and import routes into the routing tables.

```
user@host# set routing-options rib-groups Default-VPN-to-TC1_VPN  import-rib
[ Default_VPN.inet.0 TC1_VPN.inet.0 ]
```

3. Configure AppQoE as service. You must configure AppQoE as service for host inbound traffic for a desired zone.

```
user@host#  set security zones security-zone untrust1 host-inbound-traffic system-services
appqoe
```

4. Create the APBR profile and define the rules.

```
user@host# set security advance-policy-based-routing profile apbr1 rule rule1 match dynamic-
application junos:GOOGLE
user@host# set security advance-policy-based-routing profile apbr1 rule rule1 match dynamic-
application junos:YAHOO
user@host# set security advance-policy-based-routing profile apbr1 rule rule1 match dynamic-
application-group junos:web
user@host# set security advance-policy-based-routing profile apbr1 rule rule1 then routing-
instance TC1_VPN
user@host# set security advance-policy-based-routing profile apbr1 rule rule1 then sla-rule
sla1
```

5. Configure active probe parameters.

```
user@host# set security advance-policy-based-routing active-probe-params probe1 settings
data-fill juniper
user@host# set security advance-policy-based-routing active-probe-params probe1 settings
data-size 100
user@host# set security advance-policy-based-routing active-probe-params probe1 settings
probe-interval 30
user@host# set security advance-policy-based-routing active-probe-params probe1 settings
probe-count 30
user@host# set security advance-policy-based-routing active-probe-params probe1 settings
```

```
burst-size 1
user@host# set security advance-policy-based-routing active-probe-params probe1 settings sla-
export-interval 60
user@host# set security advance-policy-based-routing active-probe-params probe1 settings
dscp-code-points 000110
```

6. Configure metrics profile.

```
user@host# set security advance-policy-based-routing metrics-profile metric1 sla-threshold
delay-round-trip 120000
user@host# set security advance-policy-based-routing metrics-profile metric1 sla-threshold
jitter 21000
user@host# set security advance-policy-based-routing metrics-profile metric1 sla-threshold
jitter-type egress-jitter
user@host# set security advance-policy-based-routing metrics-profile metric1 sla-threshold
packet-loss 2
```

7. Configure underlay interfaces.

   if link-type is not configured under the underlay interfaces option, the default link-type IP is used and default link-speed of 1000 Mbps is considered.

```
user@host# set security advance-policy-based-routing underlay-interface ge-0/0/2 unit 0 link-
type MPLS
user@host# set security advance-policy-based-routing underlay-interface ge-0/0/2 unit 0
speed 800
user@host# set security advance-policy-based-routing underlay-interface ge-0/0/3 unit 0 link-
type MPLS
user@host# set security advance-policy-based-routing underlay-interface ge-0/0/3 unit 0
speed 500
```

8. Configure overlay paths.

```
user@host# set security advance-policy-based-routing overlay-path overlay-path1 tunnel-path
local ip-address 10.40.1.2
user@host# set security advance-policy-based-routing overlay-path overlay-path1 tunnel-path
remote ip-address 10.40.1.1
user@host# set security advance-policy-based-routing overlay-path overlay-path1 probe-path
local ip-address 10.40.1.2
user@host# set security advance-policy-based-routing overlay-path overlay-path1 probe-path
```

```
remote ip-address 10.40.1.1
user@host# set security advance-policy-based-routing overlay-path overlay-path2 tunnel-path
local ip-address 10.41.1.2
user@host# set security advance-policy-based-routing overlay-path overlay-path2 tunnel-path
remote ip-address 10.41.1.1
user@host# set security advance-policy-based-routing overlay-path overlay-path2 probe-path
local ip-address 10.41.1.2
user@host# set security advance-policy-based-routing overlay-path overlay-path2 probe-path
remote ip-address 10.41.1.1
user@host# set security advance-policy-based-routing overlay-path overlay-path3 tunnel-path
local ip-address 10.42.1.2
user@host# set security advance-policy-based-routing overlay-path overlay-path3 tunnel-path
remote ip-address 10.42.1.1
user@host# set security advance-policy-based-routing overlay-path overlay-path3 probe-path
local ip-address 10.42.1.2
user@host# set security advance-policy-based-routing overlay-path overlay-path3 probe-path
remote ip-address 10.42.1.1
```

9. Configure destination path groups.

```
user@host# set security advance-policy-based-routing destination-path-group site1 probe-
routing-instance transit
user@host# set security advance-policy-based-routing destination-path-group site1 overlay-
path overlay-path1
user@host# set security advance-policy-based-routing destination-path-group site1 overlay-
path overlay-path2
user@host# set security advance-policy-based-routing destination-path-group site1 overlay-
path overlay-path3
```

10. Configure multipath rule.

```
user@host# set security advance-policy-based-routing multipath-rule multi1 bandwidth-limit 60
user@host# set security advance-policy-based-routing multipath-rule multi1 application
junos:YAHOO
user@host# set security advance-policy-based-routing multipath-rule multi1 application
junos:GOOGLE
user@host# set security advance-policy-based-routing multipath-rule multi1 application-group
junos:web
user@host# set security advance-policy-based-routing multipath-rule multi1 link-type MPLS
user@host# set security advance-policy-based-routing multipath-rule multi1 link-type IP
user@host# set security advance-policy-based-routing multipath-rule multi1 max-time-to-wait
```

```
30
user@host# set security advance-policy-based-routing multipath-rule multi1 number-of-paths 2
```

11. Configure SLA rule.

```
user@host# set security advance-policy-based-routing sla-rule sla1 switch-idle-time 40
user@host# set security advance-policy-based-routing sla-rule sla1 metrics-profile metric1
user@host# set security advance-policy-based-routing sla-rule sla1 active-probe-params probe1
user@host# set security advance-policy-based-routing sla-rule sla1 passive-probe-params
sampling-percentage 25
user@host# set security advance-policy-based-routing sla-rule sla1 passive-probe-params
violation-count 2
user@host# set security advance-policy-based-routing sla-rule sla1 passive-probe-params
sampling-period 60000
user@host# set security advance-policy-based-routing sla-rule sla1 passive-probe-params type
book-ended
```

12. Associate an SLA rule to multipath rule.

```
user@host# set security advance-policy-based-routing sla-rule sla1 multipath-rule multi1
```

Configure Multipath Rules for Application Traffic (Device Configured to Receive Traffic))

Step-by-Step Procedure

The variables configured in this step are the same for both the sending and receiving device.

1. Configure multipath rule on the receiving device.

```
user@host# set security advance-policy-based-routing multipath-rule multi1 bandwidth-limit 60
user@host# set security advance-policy-based-routing multipath-rule multi1 application
junos:YAHOO
user@host# set security advance-policy-based-routing multipath-rule multi1 application
junos:GOOGLE
user@host# set security advance-policy-based-routing multipath-rule multi1 application-group
junos:web
user@host# set security advance-policy-based-routing multipath-rule multi1 link-type MPLS
user@host# set security advance-policy-based-routing multipath-rule multi1 link-type IP
```

Results

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

Hub-side device multipath rule configuration

```
[edit security]
user@host# show advance-policy-based-routing multipath-rule multi1
multipath-rule multi1 {
    bandwidth-limit 60;
    application [ junos:YAHOO junos:GOOGLE ];
    application-group junos:web;
    link-type [ MPLS IP ];
    number-of-paths 2;
}
```

```
[edit security]
user@host# show advance-policy-based-routing
profile apbr1 {
    rule rule1 {
        match {
            dynamic-application [ junos:GOOGLE, junos:YAHOO ];
            dynamic-application-group [ junos:web ];
        }
        then {
            routing-instance TC1_VPN;
            sla-rule {
                sla1;
            }
        }
    }
}
active-probe-params probe1 {
    settings {
        data-fill {
            juniper;
        }
        data-size {
            100;
        }
        probe-interval {
            30;
        }
        probe-count {
            30;
        }
        burst-size {
            1;
        }
        sla-export-interval {
            60;
        }
        dscp-code-points {
            000110;
        }
    }
}
metrics-profile metric1 {
    sla-threshold {
        delay-round-trip {
            120000;
        }
        jitter {
            21000;
        }
        jitter-type {
            egress-jitter;
        }
        packet-loss {
            2;
        }
    }
}
underlay-interface ge-0/0/2 {
    unit 0 {
        link-type MPLS;
        speed 800;
    }
}
underlay-interface ge-0/0/3 {
    unit 0 {
        link-type MPLS;
        speed 500;
    }
}
overlay-path overlay-path1 {
    tunnel-path {
        local {
            ip-address {
                10.40.1.2;
            }
        }
        remote {
            ip-address {
                10.40.1.1;
            }
        }
    }
    probe-path {
        local {
            ip-address {
                10.40.1.2;
            }
        }
        remote {
            ip-address {
                10.40.1.1;
            }
        }
    }
}
overlay-path overlay-path2 {
    tunnel-path {
        local {
            ip-address {
                10.41.1.2;
            }
        }
        remote {
            ip-address {
                10.41.1.1;
            }
        }
    }
    probe-path {
        local {
            ip-address {
                10.41.1.2;
            }
        }
        remote {
            ip-address {
                10.41.1.1;
            }
        }
    }
}
overlay-path overlay-path3 {
    tunnel-path {
        local {
            ip-address {
                10.42.1.2;
            }
        }
        remote {
            ip-address {
                10.42.1.1;
            }
        }
    }
    probe-path {
        local {
            ip-address {
                10.42.1.2;
            }
        }
        remote {
            ip-address {
                10.42.1.1;
            }
        }
    }
}
destination-path-group site1 {
    probe-routing-instance {
        transit;
    }
    overlay-path overlay-path1;
    overlay-path overlay-path2;
    overlay-path overlay-path3;
}
sla-rule sla1 {
    switch-idle-time {
        40;
    }
    metrics-profile {
        metric1;
    }
    active-probe-params {
        probe1;
    }
    passive-probe-params {
        sampling-percentage {
            25;
        }
        violation-count {
            2;
        }
        sampling-period {
            60000;
        }
        type {
            book-ended;
        }
    }
    multipath-rule {
        multi1;
    }
}
multipath-rule multi1 {
    bandwidth-limit 60;
    application [ junos:YAHOO junos:GOOGLE ];
    application-group junos:web;
    link-type [ MPLS IP ];
    number-of-paths 2;
}
```

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

Verification

IN THIS SECTION

Displaying Multipath Rule Status

Purpose

Display the details of the multipath rule on the device configured to send traffic.

Action

From operational mode, enter the `show security advance-policy-based-routing multipath rule` command.

```
user@host> show security advance-policy-based-routing multipath rule multi1
Multipath Rule Status:
  Multipath Rule Information:
    Multipath rule name                multi1
    Multipath rule type                Packet-Copy
    Multipath rule state               Active
    Configured number of paths         2
    Configured application groups      junos:web
    Configured applications            junos:GOOGLE, junos:YAHOO
  Path Group Information:
    Total path groups : 1
    Path-Group-Id  State               Avl-Num-Paths
    1              Active              3
  Receiver Information:
    Path Groups Information:
      Total receiver path groups : 1
      Path-Group-Id : 1, Avg-Pkt-Loss(%) : 0, Avg-Ingress-Jitter(us) : 171
        Path Information:
```

```
Dst-IP       Pkts-Rcvd     Pkt-Loss(%)    Ingress-Jitter(us)       Reduction-Pkt-Loss(%)    Reduction-Ingress-Jitter(us)

  10.40.1.2       2442             0                      165           0                                        -6

  10.41.1.2       2442             0                      158           0

-13


Cos Q Statistics:
          Total receiver cos queues: 8
          COS-Q-Id       Pkts-Rcvd      Out-Of-Seq-Drop
          0              4884           2442
          1              0              0
          2              0              0
          3              0              0
          4              0              0
          5              0              0
          6              0              0
          7              0              0
```

Meaning

The command output displays the multipath rule details.

Display Multipath Rule Statistics for An Application

Purpose

Display the details of the application traffic on the device configured to receive traffic

Action

From operational mode, enter the `show security advance-policy-based-routing multipath rule` *`rule-name`* `application` *`application-name`* command.

```
user@host> show security advance-policy-based-routing multipath rule multi1 application
junos:YAHOO
Multipath Rule Status:
  Multipath Rule Information:
    Multipath rule name                 multi1
    Multipath rule type                 Packet-Copy
    Multipath rule state                Active
    Configured number of paths         2
    Configured applications            junos:YAHOO
  Sender Information:
    Statistics:
      Current Sessions                   0
      Ignored Sessions                   1
      Applications Matched               1
      Applications Switched              0
      Stopped due to Bandwidth Limit     0
      Packets in path inactive state     0
      Packets in path active state       627
      Midstream Packets Ignored          0
      Total Packets Processed            627
      Total Packets Copied               627
```

Meaning

The command output displays the multipath rule for the application.

Displaying Multipath Rule Policies

Purpose

Display the details of the multipath rule on the device configured to send traffic.

Action

From operational mode, enter the `show security advance-policy-based-routing multipath rule` command.

```
user@host> show security advance-policy-based-routing multipath policy statistics application
junos:YAHOO multipath-name multi1 profile apbr1 rule rule1 zone trust
Sender Information:
    Statistics:
      Current Sessions                   0
      Ignored Sessions                   0
      Applications Matched               1
      Applications Switched              0
      Stopped due to Bandwidth Limit     0
      Packets in path inactive state     26
      Packets in path active state       2416
      Less than Configured Paths         0
```

```
        Midstream Packets Ignored          0
        Total Packets Processed            2442
        Total Packets Copied               2442
```

Meaning

The command output displays the details on the traffic handled with multipath rule applied.

Displaying Multipath Rule Status

Purpose

Display the details of the multipath rule on the device configured to receive traffic

Action

From operational mode, enter the `show security advance-policy-based-routing multipath rule` command.

```
user@host> show security advance-policy-based-routing multipath rule multi1
Multipath Rule Status:
  Multipath Rule Information:
    Multipath rule name                 multi1
    Multipath rule type                 Packet-Copy
    Multipath rule state                Active
    Configured number of paths          2
    Configured application groups       junos:web
    Configured applications             junos:GOOGLE, junos:YAHOO
  Path Group Information:
    Total path groups : 1
    Path-Group-Id  State                Avl-Num-Paths
    1              Active               3
  Receiver Information:
    Path Groups Information:
      Total receiver path groups : 1
      Path-Group-Id : 1, Avg-Pkt-Loss(%) : 0, Avg-Ingress-Jitter(us) : 171
        Path Information:


Dst-IP       Pkts-Rcvd     Pkt-Loss(%)    Ingress-Jitter(us)       Reduction-Pkt-Loss(%)    Reduction-Ingress-Jitter(us)

  10.40.1.1       2442             0                      165          0                                        -6
  10.41.1.1       2442             0                      158           0
-13


Cos Q Statistics:
          Total receiver cos queues: 8
          COS-Q-Id       Pkts-Rcvd      Out-Of-Seq-Drop
          0              4884           2442
          1              0              0
          2              0              0
          3              0              0
          4              0              0
          5              0              0
          6              0              0
          7              0              0
```

Meaning

Output displays details related to multipath rule.

Release History Table

Release	Description
21.2R1	AMR support for the traffic in reverse direction
21.2R1	AMR support for the queuing mechanism for out-of-order packets at the receiving device
21.2R1	Association of AMR rules and SLA rules with advanced policy-based routing (APBR) rule in a APBR profile
21.2R1	Link selection option that includes overlay-interfaces such as generic routing encapsulation (GRE) and secure tunnel (st)
21.2R1	Enabling of AMR in one of the two modes—SLA violation mode or standalone mode
21.2R1	AMR support for IPv6 traffic
21.2R1	Support for AMR over IPsec and Generic Routing Encapsulation (GRE) sessions
15.1X49-D170	SRX Series Firewalls in chassis cluster mode support application-based multipath routing starting in Junos OS Release 19.2R1 and Junos OS Release 15.1X49-D170.

15.1X49-D160	SRX Series Firewalls support application-based multipath routing starting in Junos OS Release 15.1X49-D160 (standalone mode).

4

CHAPTER

SSL Proxy

SSL Proxy

IN THIS SECTION

SSL proxy acts as an intermediary, performing SSL encryption and decryption between the client and the server. Better visibility into application usage can be made available when SSL forward proxy is enabled.

SSL Proxy Overview

IN THIS SECTION

SSL proxy is supported on SRX Series Firewalls only.

Secure Sockets Layer (SSL) is an application-level protocol that provides encryption technology for the Internet. SSL, also called Transport Layer Security (TLS), ensures the secure transmission of data between a client and a server through a combination of privacy, authentication, confidentiality, and data integrity. SSL relies on certificates and private-public key exchange pairs for this level of security.

SSL proxy is transparent proxy that performs SSL encryption and decryption between the client and the server.

How Does SSL Proxy Work?

SSL proxy provides secure transmission of data between a client and a server through a combination of following:

- Authentication-Server authentication guards against fraudulent transmissions by enabling a Web browser to validate the identity of a webserver.
- Confidentiality - SSL enforces confidentiality by encrypting data to prevent unauthorized users from eavesdropping on electronic communications; thus ensures privacy of communications.
- Integrity- Message integrity ensures that the contents of a communication are not tampered.

SRX Series Firewall acting as SSL proxy manages SSL connections between the client at one end and the server at the other end and performs following actions:

- SSL session between client and SRX Series- Terminates an SSL connection from a client, when the SSL sessions are initiated from the client to the server. The SRX Series Firewall decrypts the traffic, inspect it for attacks (both directions), and initiates the connection on the clients' behalf out to the server.
- SSL session between server and SRX Series - Terminates an SSL connection from a server, when the SSL sessions are initiated from the external server to local server. The SRX Series Firewall receives clear text from the client, and encrypts and transmits the data as ciphertext to the SSL server. On the other side, the SRX Series decrypts the traffic from the SSL server, inspects it for attacks, and sends the data to the client as clear text.
- Allows inspection of encrypted traffic.

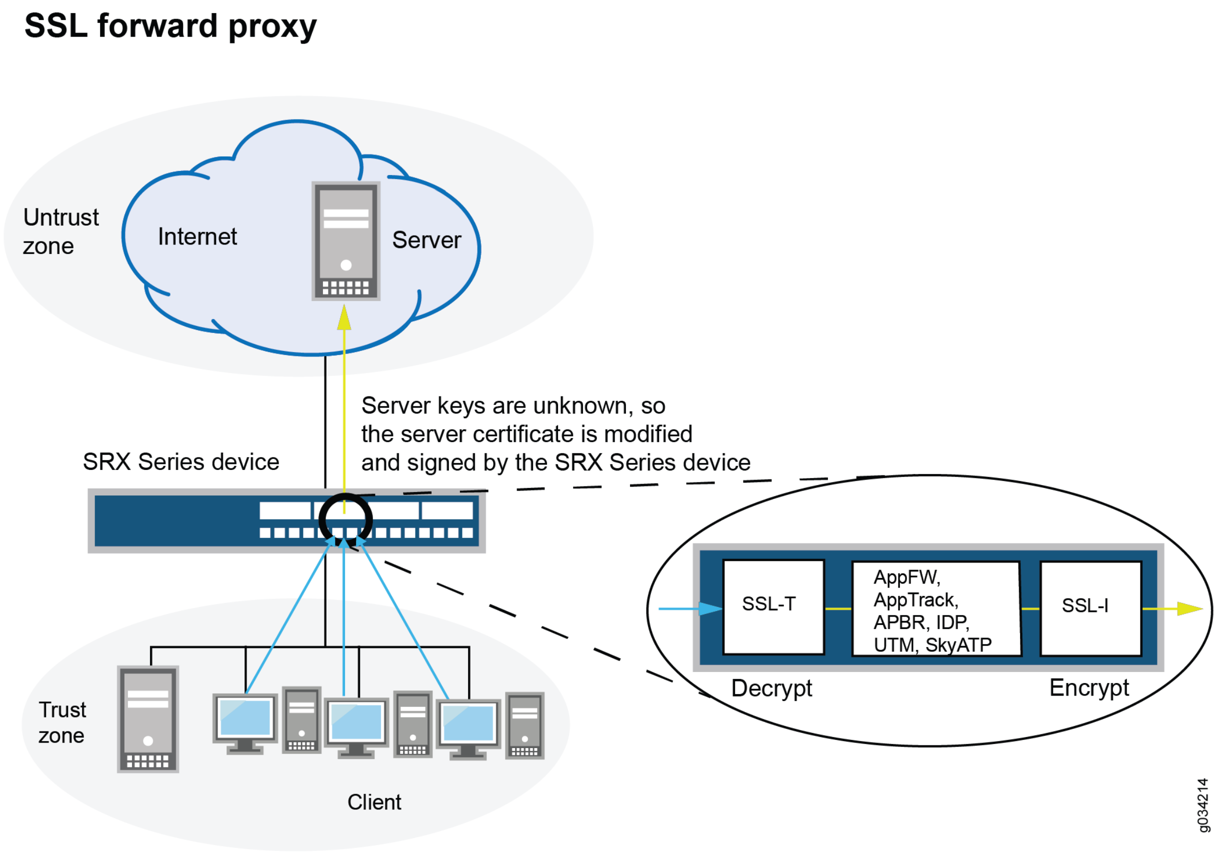
SSL proxy server ensures secure transmission of data with encryption technology. SSL relies on certificates and private-public key exchange pairs to provide the secure communication. For more information, see "SSL Certificates" on page 355.

To establish and maintain an SSL session between the SRX Series Firewall and its client/server, the SRX Series Firewall applies security policy to the traffic that it receives. When the traffic match the security policy criteria, SSL proxy is enabled as an application service within a security policy.

SSL Proxy with Application Security Services

Figure 14 on page 352 shows how SSL proxy works on an encrypted payload.

Figure 14: SSL Proxy on an Encrypted Payload


SSL forward proxy
Untrust zone
Internet
Server
Server keys are unknown, so the server certificate is modified and signed by the SRX Series device
SRX Series device
SSL-T
AppFW, AppTrack, APBR, IDP, UTM, SkyATP
SSL-I
Decrypt
Encrypt
Trust zone
Client
g034214

When Advanced Security services such as application firewall (AppFW), Intrusion Detection and Prevention (IDP), application tracking (AppTrack), Content Security, and ATP Cloud is configured, the SSL proxy acts as an SSL server by terminating the SSL session from the client and establishing a new SSL session to the server. The SRX Series Firewall decrypts and then reencrypts all SSL proxy traffic.

IDP, AppFW, AppTracking, advanced policy-based routing (APBR), Content Security, ATP Cloud, and ICAP service redirect can use the decrypted content from SSL proxy. If none of these services are configured, then SSL proxy services are bypassed even if an SSL proxy profile is attached to a firewall policy.

Types of SSL Proxy

SSL proxy is a transparent proxy that performs SSL encryption and decryption between the client and the server. SRX acts as the server from the client's perspective and it acts as the client from the server's perspective. On SRX Series Firewalls, client protection (forward proxy) and server protection (reverse proxy) are supported using same echo system SSL-T-SSL [terminator on the client side] and SSL-I-SSL [initiator on the server side]).

SRX Series Firewall support following types of SSL proxy:

- Client-protection SSL proxy also known as forward proxy—The SRX Series Firewall resides between the internal client and outside server. Proxying outbound session, that is, locally initiated SSL session to the Internet. It decrypts and inspects traffic from internal users to the web.
- Server-protection SSL proxy also known as reverse proxy—The SRX Series Firewall resides between the internal server and outside client. Proxying inbound session, that is, externally initiated SSL sessions from the Internet to the local server.

For more information on SSL forward proxy and reverse proxy, see "Configuring SSL Proxy" on page 386.

Supported SSL Protocols

The following SSL protocols are supported on SRX Series Firewalls for SSL initiation and termination service:

- TLS version 1.0—Provides authentication and secure communications between communicating applications.
- TLS version 1.1—This enhanced version of TLS provides protection against cipher block chaining (CBC) attacks.
- TLS version 1.2 — This enhanced version of TLS provides improved flexibility for negotiation of cryptographic algorithms.
- TLS version 1.3 — This enhanced version of TLS provides improved security and better performance.

Starting with Junos OS Release 15.1X49-D30 and Junos OS Release 17.3R1, TLS version 1.1 and TLS version 1.2 protocols are supported on SRX Series Firewalls along with TLS version 1.0.

Starting with Junos OS Release 15.1X49-D20 and Junos OS Release 17.3R1, the SSL protocol 3.0 (SSLv3) support is deprecated.

Starting in Junos OS Release 21.2R1, on SRX Series Firewalls, SSL proxy supports TLS version 1.3.

When you use TLS 1.3, the SRX Series Firewall supports the secp256r1 group for key-exchange for establishing connection with the server. If the server supports only secp384r1, then the connection will be terminated.

Benefits of SSL Proxy

- Decrypts SSL traffic to obtain granular application information and enable you to apply advanced security services protection and detect threats.
- Enforces the use of strong protocols and ciphers by the client and the server.

- Provides visibility and protection against threats embedded in SSL encrypted traffic.
- Controls what needs to be decrypted by using Selective SSL Proxy.

Logical Systems Support

It is possible to enable SSL proxy on firewall policies that are configured using logical systems; however, note the following limitations:

- The “services” category is currently not supported in logical systems configuration. Because SSL proxy is under “services,” you cannot configure SSL proxy profiles on a per-logical-system basis.
- Because proxy profiles configured at a global level (within “services ssl proxy”) are visible across logical system configurations, it is possible to configure proxy profiles at a global level and then attach them to the firewall policies of one or more logical systems.

Limitations

On all SRX Series Firewalls, the current SSL proxy implementation has the following connectivity limitations:

- The SSLv3.0 protocol support is deprecated.
- The SSLv2 protocol is not supported. SSL sessions using SSLv2 are dropped.
- Only X.509v3 certificate is supported.
- Client authentication of SSL handshake is not supported.
- SSL sessions where client certificate authentication is mandatory are dropped.
- SSL sessions where renegotiation is requested are dropped.

On SRX Series Firewalls, for a particular session, the SSL proxy is only enabled if a relevant feature related to SSL traffic is also enabled. Features that are related to SSL traffic are IDP, application identification, application firewall, application tracking, advanced policy-based routing, Content Security, ATP Cloud, and ICAP redirect service. If none of these features are active on a session, the SSL proxy bypasses the session and logs are not generated in this scenario.

SEE ALSO

SSL Certificates | 355

Configuring SSL Proxy | 386

Unified Policies for SSL Proxy | 409

SSL Certificates

IN THIS SECTION

SRX Series Firewall acting as SSL proxy manages SSL connections between the client at one end and the server at the other end. SSL proxy server ensures secure transmission of data with encryption technology. SSL relies on certificates and private-public key exchange pairs to provide the secure communication. In this topic, you learn about how to generate and install SSL certificate on your security device for SSL connections.

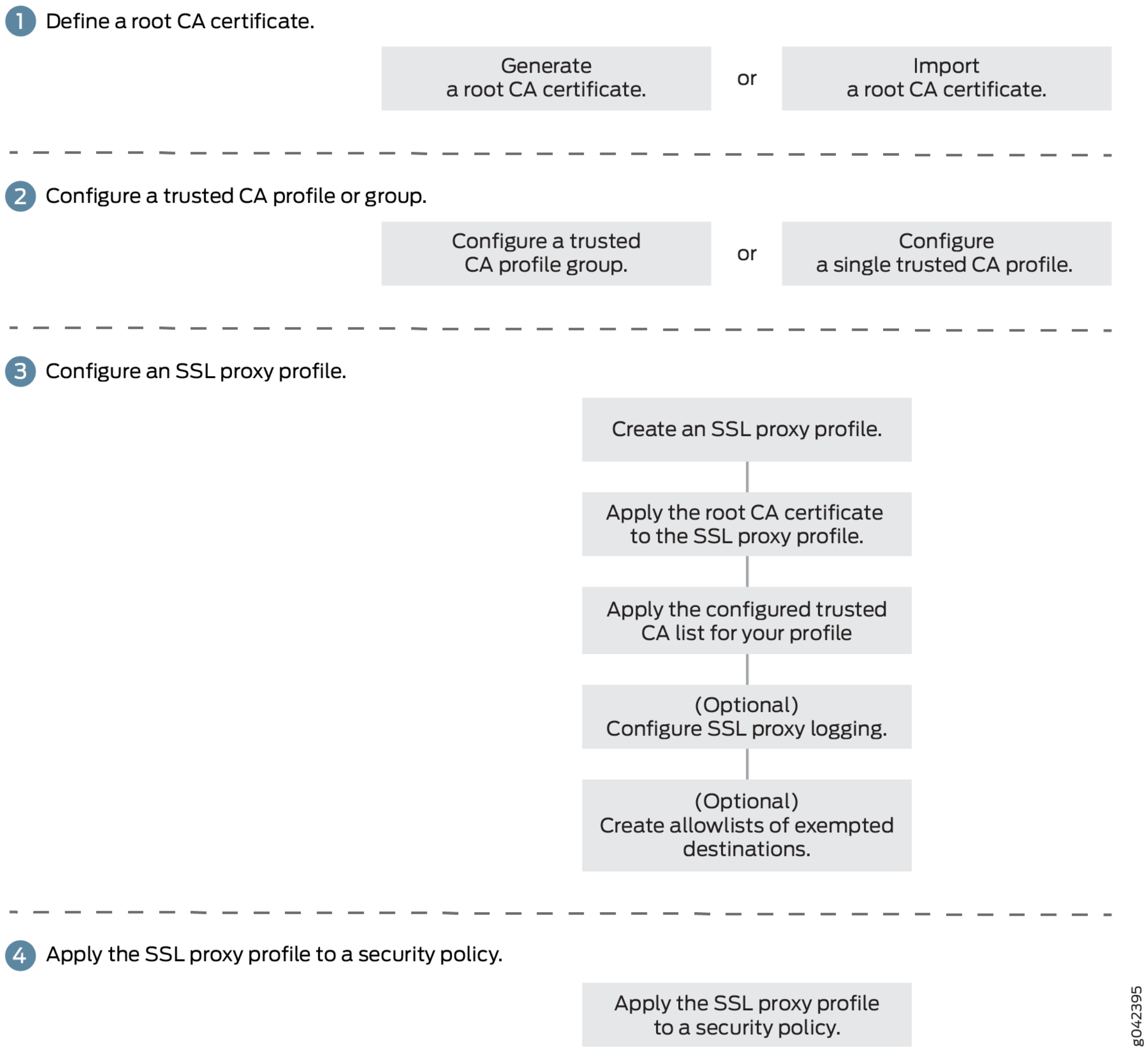
Configuring and Loading SSL Certificates

Figure 15 on page 356 displays an overview of how SSL proxy is configured. Configuring SSL proxy includes:

- Configuring the root CA certificate
- Loading a CA profile group

- Configure SSL proxy profile and associate root CA certificate and CA profile group
- Applying an SSL proxy profile to a security policy

Figure 15: SSL Proxy Configuration Overview


1 Define a root CA certificate.
Generate a root CA certificate.
or
Import a root CA certificate.
2 Configure a trusted CA profile or group.
Configure a trusted CA profile group.
or
Configure a single trusted CA profile.
3 Configure an SSL proxy profile.
Create an SSL proxy profile.
Apply the root CA certificate to the SSL proxy profile.
Apply the configured trusted CA list for your profile
(Optional) Configure SSL proxy logging.
(Optional) Create allowlists of exempted destinations.
4 Apply the SSL proxy profile to a security policy.
Apply the SSL proxy profile to a security policy.
g042395

Lets discuss these procedures in detail in the following sections:

Configuring a Root CA Certificate

IN THIS SECTION

A CA can issue multiple certificates in the form of a tree structure. A root certificate is the topmost certificate of the tree, the private key of which is used to *sign* other certificates. All certificates immediately below the root certificate inherit the signature or trustworthiness of the root certificate. This is somewhat like the *notarizing* of an identity.

To configure a root CA certificate you must

1. Obtaining a root CA certificate (by either or importing one)
   - You can generate a root CA certificate using Junos OS CLI on an SRX Series Firewall.
   - Obtain a certificate from an External CA (not covered in this topic)
2. Applying root CA to an SSL proxy profile.

Generate a Root CA Certificate with CLI

To define a self-signed certificate in CLI, you must provide the following details:

- Certificate identifier (generated in the previous step)
- Fully qualified domain name (FQDN) for the certificate
- e-mail address of the entity owning the certificate
- Common name and the organization involved

Generate a root CA certificate using the Junos OS CLI:

1. From operational mode, generate a PKI public/private key pair for a local digital certificate.

   Example:

   ```
   user@host>request security pki generate-key-pair certificate-id SECURITY-cert size 2048 type
   rsa
   ```

2. Define a self-signed certificate.

Example:

```
user@host> request security pki local-certificate generate-self-signed certificate-id
SECURITY-cert domain-name labs.abc.net subject
DC=mydomain.net,L=Sunnyvale,O=Mydomain,OU=LAB,CN=SECURITY email lab@labs.abc.net add-ca-
constraint
```

By configuring the `add-ca-constraint` option, you make sure that the certificate can be used for signing other certificates.

3. From configuration mode, apply the loaded certificate as root-ca in the SSL proxy profile.

```
[edit]
user@host# set services ssl proxy profile profile-name root-ca certificate-id
```

Example:

```
[edit]
user@host# set services ssl proxy profile SECURITY-SSL-PROXY root-ca SECURITY-cert
```

4. Import the root CA as a trusted CA into client browsers. This is required for the client browsers to trust the certificates signed by the SRX Series Firewall. See "Importing a Root CA Certificate into a Browser" on page 361.

Configuring a Trusted CA Profile Group

IN THIS SECTION

The CA profile defines the certificate information for authentication. It includes the public key that SSL proxy uses when generating a new certificate. Junos OS allows you to create a group of CA profiles and load multiple certificates in one action, view information about all certificates in a group, and delete unwanted CA groups. When a connection is initiated, the connecting device (such as a Web browser) checks whether the certificate is issued by a trusted CA. Without these certificates, browsers cannot validate the identity of most websites and mark them as untrusted sites.

Configuring a trusted CA profile group includes following steps:

- Obtaining a list of trusted CA certificates. You can obtain trusted CA certificates using one of the following methods:

  - Junos OS provides a default list of trusted CA certificates as a PEM file (for example, **trusted_CA.pem**). After you download the Junos OS package, the default certificates are available on your system.

    From operational mode, load the default trusted CA certificates (the group name identifies the CA profile group):

    ```
    user@host> request security pki ca-certificate ca-profile-group load ca-group-name group-name filename default
    ```

    Example:

    ```
    user@host> request security pki ca-certificate ca-profile-group load ca-group-name
    SECURITY-CA-GROUP filename default
    ```

    The SRX Series Firewall support dynamic update of default trusted CA certificates. The latest default trusted CA certificates are stored and periodically updated on Juniper CDN server (http://signatures.juniper.net/cacert). Automatic download of trusted CA bundle is enabled by default in Junos OS device. This eliminates the need for managing these certificates manually in the event of compromise. To use this feature, see Dynamic Updated of Trusted CA Certificates.

    We recommend using this method.

  - Define your own list of trusted CA certificates and import them on your system. You get the list of trusted CAs in a single PEM file (for example **IE-all.pem**) and save the PEM file in a specific location (for example, **/var/tmp**). See Knowledge Base Article KB23144.

    From operational mode, load the trusted list to the device (the group name identifies the CA profile group):

    ```
    user@host> request security pki ca-certificate ca-profile-group load ca-group-name
    SECURITY-CA-GROUP filename /var/tmp/IE-all.pem
    ```

    Example:

    ```
    user@host> request security pki ca-certificate ca-profile-group load ca-group-name
    SECURITY-CA-GROUP filename /var/tmp/custom-file.pem
    ```

  - Download the latest CA bundle list from another 3rd party such as Mozilla (https://curl.haxx.se/docs/caextract.html). The list of trusted Certificate Authority can change over time, ensure that you use the latest CA bundle.

  - Import your own trusted CA certificates using the Public Key Infrastructure (PKI). The PKI helps verify and authenticate the validity of the trusted CA certificates. You create CA profile groups that include trusted CA certificates, then import the group on your device for server authentication.

- Attaching the CA group to the SSL proxy profile.

  - Attach all CA profile groups:

    Example:

    ```
    [edit]
    user@host# set services ssl proxy profile PROFILE-1 trusted-ca all
    ```

  - Attach one CA profile group (the group name identifies the CA profile group).

    Example:

    ```
    [edit]
    user@host# set services ssl proxy profile PROFILE-1 trusted-ca orgA-ca-profile
    ```

You can easily display information about all certificates in a CA profile group:

```
user@host> show security pki ca-certificates ca-profile-group group-name
```

You can delete a CA profile group. Remember that deleting a CA profile group deletes all certificates that belong to that group:

```
user@host> clear security pki ca-certificates ca-profile-group group-name
```

WHAT'S NEXT

Now proceed with SSL proxy profile configuration and apply SSL proxy profile to security policy. See "Configuring SSL Proxy" | 386.

Importing a Root CA Certificate into a Browser

In order to have your browser or system automatically trust all certificates signed by the root CA configured in the SSL proxy profile, you must instruct your platform or browser to trust the CA root certificate.

To import a root CA certificate:

1. Generate a PEM format file for the configured root CA.

```
request security pki local-certificate export certificate-id root-ca type pem filename path/file-name.pem
```

2. Import a root CA certificate into a browser.

   From Internet Explorer (version 8.0):

   a. From the Tools menu, select **Internet Options**.

   b. On the Content tab, click **Certificates**.

   c. Select the **Trusted Root Certification Authorities** tab and click **Import**.

   d. In the Certificate Import Wizard, navigate to the required root CA certificate and select it.

   From Firefox (version 39.0):

   a. From the Tools menu, select **Options**.

   b. From the Advanced menu, select the **Certificates** tab and click **View Certificate**.

   c. In the Certificate Manager window, select the **Authorities** tab and click **Import**.

   d. Navigate to the required root CA certificate and select it.

   From Google Chrome (45.0):

   a. From the Settings menu, select **Show Advanced Settings**.

   b. From the Advanced menu, select the **Certificates** tab and click **View Certificate**.

   c. Under HTTPS/SSL, click **Manage Certificates**.

   d. In the Certificate window, select **Trusted Root Certification Authorities** and click **Import**.

   e. In the Certificate Import Wizard, navigate to the required root CA certificate and select it.

Certificate Chain Implementation

IN THIS SECTION

Starting in Junos OS Release 15.1X49-D30, SSL forward proxy supports the certificate chain implementation. Lets discuss about understanding certificate chain concepts and how to configure it on SRX Series Firewall.

- **Certificate Authority (CA)**— CA is a trusted third party responsible for validating the identities of entities (such as websites, email addresses, or companies, or individual persons) and issues a digital certificate by binding cryptographic keys. If your organization owns a CA server, then you become your own CA and use self-signed certificate.
- **Root Certificate**—A Root certificate is a certificate issued by a trusted certificate authority (CA). The root certificate is the topmost certificate of the tree, the private key of which is used to sign other certificates. All certificates immediately below the root certificate inherit the signature or trustworthiness of the root certificate. These certificates are used to establish connection between two endpoints.
- **Intermediate CA Certificate**—An intermediate CA certificate is a subordinate certificate signed by the trusted root specifically to validate an end-entity certificates.
- **Certificate Chain** - An certificate chain is the ordered list of certificates that contains the SSL certificate, intermediate certificate, and root certificate. Some certificate authorities (CAs) do not sign with their root certificate, but instead use an intermediate certificate. An intermediate CA can sign certificates on behalf of the root CA certificate. The root CA signs the intermediate certificate, forming a chain of trust.

  The intermediate certificate must be installed on the same server as the SSL certificate so that the connecting device (browsers, applications, mobile device, etc.) can trust it.

When you initiate a connection, the connecting device (such as a Web browser) checks whether the certificate is authentic and is issued by a trusted certificate authority that is embedded in the browser's trusted store.

If the SSL certificate is not from a trusted CA, then the connecting device continues to check if the SSL certificate is issued by a intermediate CA and this intermediate CA is signed by a root CA. The check continues till the root CA is found. If it finds a root CA, a secure connection is established. If it doesn't find a root CA, then the connection is dropped, and your web browser displays an error message about invalid certificate or certificate not trusted.

This example shows how to install the certificate chain to enable browsers to trust your certificate.

Requirements

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

Overview

IN THIS SECTION

In this example, you have a domain, example.domain-1, and you want to purchase a certificate from XYZ-Authority for your domain. However, XYZ -Authority is not a Root-CA and the visiting Web browser trusts only Root-CA certificate. In other words, its certificate is not directly embedded in your Web browser and therefore it is not explicitly trusted.

In this case, trust is established in the following manner using the certificate chain (of intermediate certificates).

Topology

Let's try to visualize this chain through Figure 16 on page 364. The image depicts a full certificate chain, from the root CA certificate to the end-user certificate. The chain terminates at the end-user certificate.

Figure 16: Certification Path from the Certificate Owner to the Root CA

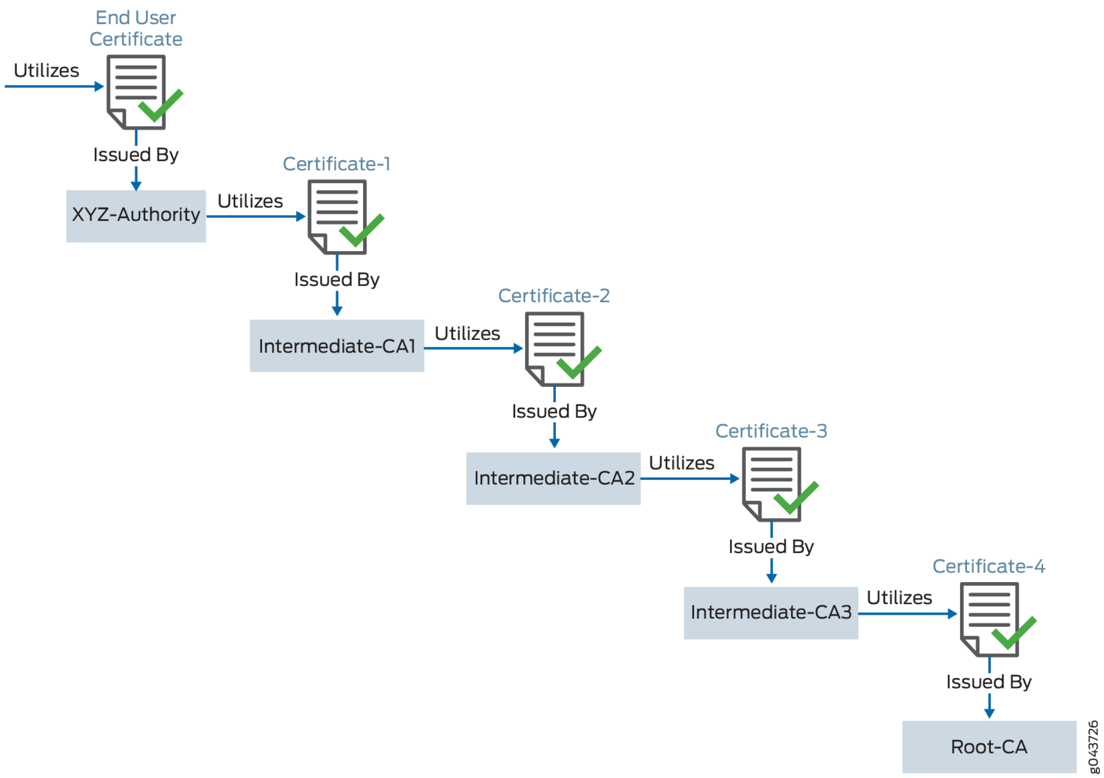


Table 25: Certificate Chaining Details

User	Uses Certificate	Signed By	Type
example.domain-1	End User Certificate	XYZ-Authority	End User Certificate. The one the one you purchase from the CA.
XYZ-Authority	Certificate-1	Intermediate CA-1	Intermediate Certificate
Intermediate CA-1	Certificate-2	Intermediate CA-2	Intermediate Certificate
Intermediate CA-2	Certificate-3	Intermediate CA-3	Intermediate Certificate

Table 25: Certificate Chaining Details *(Continued)*

User	Uses Certificate	Signed By	Type
Intermediate CA-3	Certificate-4	root-example-authority. This is a root CA.	Root Certificate Its certificate is directly embedded in your Web browser; therefore it can be explicitly trusted.

When you install your end-user certificate for the server example.domain-1, you must bundle all the intermediate certificates and install them along with your end-user certificate. The certificate chain includes all the certificates starting from Certificate-1 to Root-CA certificate. Because the web browser trusts the root CA, it also implicitly trusts all the intermediate certificates. If the SSL certificate chain is invalid or broken, your certificate will not be trusted by some devices.

NOTE:

- All certificates must be in Privacy-Enhanced Mail (PEM) format.
- When you import the concatenated certificate file into the device, the CA provides a bundle of chained certificates that must be added to the signed server certificate. The server certificate must appear before the chained certificates in the combined file.

Configuration

IN THIS SECTION

Configuring the SSL certificate chain includes the following tasks:

- Purchase an SSL certificate from a CA that includes a signing certificate and a respective key.
- Configure a trusted CA profile group.
- Load the signing certificate and the key on your device.
- Load the intermediate and root CA in public key infrastructure (PKI) memory. This certificate file contains all the required CA certificates, one after each other, in PEM format.

- Create a trusted CA profile for the intermediate or root CA certificate.
- Set up your device to use the signing certificate received from the CA by configuring and applying the SSL proxy profile to a security policy. SSL forward proxy stores this certificate chain information (CA certificate profile name) in the respective SSL profile. As a part of security policy implementation, SSL profiles having the certificate chain information and CA certificates are used.

The following components are involved in certificate chain processing:

- Administrator loads the certificate chain and the local certificate (signing certificate) into the PKI daemon certificate cache.
- The Network Security Daemon (nsd) sends a request to the PKI daemon to provide the certificate chain information for a signing certificate configured in the SSL proxy profile.

This example assumes that you have already purchased an SSL certificate from a CA.

Configuring the Certificate Chain on the Device

Step-by-Step Procedure

To configure certificate chain:

- Load the local certificate into the PKI memory.

  ```
  user@host>  request security pki local-certificate load filenamessl_proxy_ ca.crt key
  sslserver.key certificate-id ssl-inspect-ca
  ```

  The following message is displayed:

  ```
  Local certificate loaded successfully
  ```

  Note that the certificate ID will be used under the `root-ca` section in the SSL proxy profile.

- Load the intermediate or root CA certificate in the PKI memory.

  ```
  user@host> request security pki ca-certificate ca-profile-group load  ca-group-name ca-latest
  filename ca-latest.cert.pem
  ```

The CA profile includes the certificate information used for authentication. It includes the public key that SSL proxy uses when generating a new certificate.

```
Do you want to load this CA certificate? [yes,no] (no) yes

      Loading 1 certificates for group 'ca-latest'.
      ca-latest_1: Loading done.
      ca-profile-group 'ca-latest' successfully loaded
      Success[1] Skipped[0]
```

This certificate will be attached as a certificate chain.

- Attach the CA profile group to the SSL proxy profile. You can attach trusted CA one at a time or load all in one action.

```
user@host# set services ssl proxy profile ssl-profile trusted-ca all
```

- Apply the signing certificate as root-ca in the SSL proxy profile.

```
user@host# set services ssl proxy profile ssl-profile root-ca ssl-inspect-ca
```

- Create a security policy and specify the match criteria for the policy. As match criteria, specify the traffic for which you want to enable SSL proxy. This example assumes that you have already created security zones based on the requirements.

```
user@host# set security policies from-zone trust to-zone untrust policy 1 match source-address
any
user@host# set security policies from-zone trust to-zone untrust policy 1 match destination-
address any
user@host# set security policies from-zone trust to-zone untrust policy 1 match application any
user@host# set security policies from-zone trust to-zone untrust policy 1 then permit
application-services ssl-proxy profile-name ssl-profile
```

SSL forward proxy stores this certificate chain information (CA certificate profile name) into respective the SSL profile. As a part of security policy implementation, SSL profiles having the certificate chain information and CA certificates are used.

You can view the certificate chain on the connecting Web browser (that is, the client).

SEE ALSO

Example: Loading CA and Local Certificates Manually

Example: Configuring a Device for Peer Certificate Chain Validation

Ignore Server Authentication Failure

IN THIS SECTION

Server Authentication

Implicit trust between the client and the device (because the client accepts the certificate generated by the device) is an important aspect of SSL proxy. It is extremely important that server authentication is not compromised; however, in reality, self-signed certificates and certificates with anomalies are in abundance. Anomalies can include expired certificates, instances of common name not matching a domain name, and so forth.

Server authentication is governed by setting the **ignore-server-auth-failure** option in the SSL proxy profile. The results of setting this option is available in Table 26 on page 369.

```
[edit]
user@host# set services ssl proxy profile profile-name actions ignore-server-auth-failure
```

Table 26: Ignore Server Authentication Failure Option

SSL Proxy Profile Action	Results
The **ignore-server-auth-failure** option is not set (Default option)	• If authentication succeeds, a new certificate is generated by replacing the keys and changing the issuer name to the issuer name that is configured in the root CA certificate in the proxy profile. • If authentication fails, the connection is dropped.
The **ignore-server-auth-failure** option is set	• If the certificate is self-signed, a new certificate is generated by replacing the keys. The issuer name is not changed. This ensures that the client browser displays a warning that the certificate is not valid. • If the certificate has expired or if the common name does not match the domain name, a new certificate is generated by replacing the keys and changing the issuer name to SSL-PROXY: DUMMY_CERT:GENERATED DUE TO SRVR AUTH FAILURE. This ensures that the client browser displays a warning that the certificate is not valid. • We do not recommend this option for authentication, because configuring it results in websites not being authenticated at all. However, you can use this option to effectively identify the root cause for dropped SSL sessions. See Enabling Debugging and Tracing for SSL Proxy.

Client Authentication

Currently, client authentication is not supported in SSL proxy. If a server requests client authentication, a warning is issued that a certificate is not available. The warning lets the server determine whether to continue or to exit.

Certificate Revocation Lists for SSL Proxy

IN THIS SECTION

Working with the Certificate Revocation Lists for SSL Proxy

Certificate authority (CA) periodically publishes a list of revoked certificate using a certificate revocation list (CRL). The security device downloads and caches the most recently issued CRL. The CRL contains the list of digital certificates with serial numbers that have been canceled before their expiration date.

CA revokes the issued certificate if there is any chance that the certificate is compromised. Some other reasons for revoking a certificate are:

- Unspecified (no particular reason is given).
- Private key associated with the certificate or CA that issued the certificate was compromised.
- The owner of the certificate is no longer affiliated with the issuer of the certificate
- Another certificate replaces the original certificate.
- The CA that issued the certificate has ceased to operate.
- The certificate is on hold pending further action. It is treated as revoked but might be accepted in the future.

When a participating device uses a digital certificate, it checks the certificate signature and validity. By default, CRL verification is enabled on SSL proxy profile.

Starting with Junos OS Release 15.1X49-D30 and Junos OS Release 17.3R1, SRX Series Firewalls support certificate revocation list (CRL). CRL validation on SRX Series Firewall involves checking for the revoked certificates from servers.

On SRX Series Firewall, the certificate revocation checking is enabled by default for SSL proxy profile. You can enable or disable the CRL validation to meet your specific security requirements.

- Disable CRL verification.

```
[edit]
user@host# set services ssl proxy profile profile-name actions crl disable
```

- Re-enable CRL verification.

```
[edit]
user@host# delete services ssl proxy profile profile-name  actions crl disable
```

You can allow or drop the sessions when a CRL information is not available for reasons such as failed CRL download or unavailability of the CRL path in the root or intermediate certificate.

- Allow the sessions when CRL information is not available.

```
[edit]
user@host# set services ssl proxy profile profile-name actions crl if-not-present allow
```

- Drop the sessions when CRL information is not available.

```
[edit]
user@host# set services ssl proxy profile profile-name actions crl if-not-present drop
```

- Configure an SRX Series Firewall to accept a certificate without a reliable confirmation available on the revocation status and allow the sessions when a certificate is revoked and the revocation reason is on hold.

```
[edit]
user@host# set services ssl proxy profile profile-name actions crl ignore-hold-instruction-code
```

RELATED DOCUMENTATION

Understanding Online Certificate Status Protocol and Certificate Revocation Lists

SSL Performance Enhancements

IN THIS SECTION

SSL performance enhancement on SRX Series Firewall includes following features:

Optimizing the SSL Performance

The SSL/TLS handshake is a CPU-intensive process. Since SSL/TLS is the most widely used security protocol on the web, it’s performance results in significant impact on the web performance.

Starting from Junos OS Release 15.1X49-D120, you can use the following options for optimizing the performance:

- Use optimized RSA key exchanges
- Use Authenticated Encryption with Associated Data (AEAD)—AES128-CBC-SHA, AES256-CBC-SHA
- Maintaining certificate cache—Certificate cache stores the interdicted server certificate along with the server certificate details. During SSL/TLS handshake, SSL proxy can present the cached interdicted certificate to client instead of generating the new interdicted certificate.

Improving the SSL performance results in improved website performance without compromising security and maximized user experience.

You can optionally configure the following settings for your certificate cache. However, we recommend retaining the default values.

Example:

- (Optional) Set the certificate cache timeout value (example- 300 seconds) .

  ```
  [edit]
   user@host# set services ssl proxy global-config certificate-cache-timeout 300
  ```

  In this example, the certificate cache stores the certificate details for 300 seconds. The default timeout value is 600 seconds.

- (Optional) Disable the certificate cache.

  ```
  [edit]
   user@host# set services ssl proxy global-config disable-cert-cache
  ```

  When you disable certificate cache, the device allows the SSL full handshake for a new connection. By default certificate cache is enabled.

- (Optional) Invalidate the existing certificate cache.

  ```
  [edit]
   user@host# set services ssl proxy global-config invalidate-cache-on-crl-update
  ```

  In this example, the device invalidates the existing certificate cache when certificate revocation list (CRL) is updated. By default, invalidate certificate cache on CRL update is disabled.

Session Resumption

SSL session resumption provides a mechanism to resume a previous session using already negotiated session IDs. Session resumption saves the client and server the computational overhead of a complete SSL handshake and generation of new primary keys.

Session resumption shortens the handshake process and accelerates SSL transactions resulting in improved performance while maintaining appropriate level of security.

TLS 1.2 supports session resumption with session identifiers and session tickets mechanisms. An SSL session resumption includes the following steps:

- A session caching mechanism caches session information, such as the pre-master secret key and agreed-upon ciphers for both the client and server.
- Session ID identifies the cached information.
- In subsequent connections both parties agree to use the session ID to retrieve the information rather than create a pre-master secret key.

Starting in Junos OS Release 22.1R1, TLS 1.3 supports session resumption using a pre-shared key (PSK) in SSL proxy. Session resumption using PSK allows resuming a session with a previously established shared secret key to reduce SSL handshake overhead.

PSK is a unique encryption key derived from the initial TLS handshake. After a successful TLS handshake, the server sends a PSK identity to the client. The client uses that PSK identity in the future handshakes to negotiate the use of associated PSK to resume the session.

An SSL session resumption with TLS1.3 includes the following steps:

1. After an initial TLS handshake, the server sends a new Session Ticket message to the client, which contains the PSK identity (encrypted copy of the PSK).
2. Next time, when the client attempts to resume the session, it sends the same Session Ticket to the server in the Hello message.
3. The server decrypts the message, identifies the PSK, and retrieves the session information from its cache to resume the session.

An SSL-I (SSL initiation) uses PSK with ECDHE key exchange mode, and SSL-T (SSL termination) uses PSK and PSK with ECDHE exchange modes.

SSL proxy session supports interoperability between TLS1.3 and TLS1.2 and earlier versions on two ends of a connection for session resumption.

By default, session resumption is enabled for SSL proxy. You can clear the session resumption or re-enable as per your requirements.

- To clear the session resumption:

```
[edit]
user@host# set services ssl proxy profile ssl actions disable-session-resumption
```

- To re-enable session resumption:

```
[edit]
user@host# delete services ssl proxy profile ssl actions disable-session-resumption
```

Use the following command to configure the session timeout value.

```
[edit]
user@host# set services ssl proxy global-config session-cache-timeout session-cache-timeout
```

You can configure the timeout value between 300 seconds to 86400 seconds. The default value is 86400 seconds (24 hours).

Session Renegotiation

The SRX Series Firewall support session renegotiation. After a session is created and SSL tunnel transport is established, a change in SSL parameters requires renegotiation. SSL proxy supports both secure (RFC 5746) and nonsecure (TLS v1.0, TLS v1.1, and TLS v1.2) renegotiation. When session resumption is enabled, session renegotiation is useful in the following situations:

- Cipher keys need to be refreshed after a prolonged SSL session.
- Stronger ciphers need to be applied for a more secure connection.

If you modify the SSL proxy profile by changing a certificate, or cipher strength, or trusted CA list, then the system flushes the cache entries when you commit the modified policy. In this case, a full handshake is required to establish the new SSL parameters. (There is no impact to non-SSL sessions.)

If the SSL proxy profile is not altered, cache entries corresponding to that profile are not flushed and the session continues.

Negotiating StartTLS for SMTP Sessions

The StartTLS enables an SMTP session from an initial plain TCP connection to a more secure TLS connection. StartTLS allows SMTP session between the server and client over the TLS layer after successful negotiation between the peers. StartTLS upgrades an existing insecure plain SMTP connection to a secure SSL/TLS connection.

You can set threshold that allow you to decide how long to wait before ignoring the the session if StartTLS is not received from the client.

You can configure the following options:

- byte-threshold—Minimum bytes of unencrypted session allowed before ignoring the session if StartTLS is not received from the client. Range 100 to 300.
- packet-threshold—Number of unencrypted packets allowed in client-to-server direction before ignoring the session if StartTLS is not received from the client. Range 1 to 15.

Example:

```
[edit]
user@host# set services ssl proxy global-config mail-threshold byte-threshold 100
```

In this exaple, SSL proxy allows 100 bytes of plain (unencrypted) SMTP traffic. After reaching 100 bytes, it ignores the session if StartTLS is not received from the client.

```
[edit]
user@host# set services ssl proxy global-config mail-threshold packet-threshold 2
```

In this example, SSL proxy allows 2 packets of plain (unencrypted) SMTP traffic. After reaching 2 packets, it ignores the session if StartTLS is not received from the client.

Dynamic Resolution of Domain Names

The IP addresses associated with domain names are dynamic and can change at any time. Whenever a domain IP address changes, it is propagated to the SSL proxy configuration (similar to what is done in the firewall policy configuration).

Release History Table

Release	Description
15.1X49-D30	Starting with Junos OS Release 15.1X49-D30 and Junos OS Release 17.3R1, SRX Series Firewalls support certificate revocation list (CRL).

RELATED DOCUMENTATION

SSL Proxy | 350
Cipher Suites for SSL Proxy | 376
ICAP Service Redirect | 422
SSL Decryption Mirroring | 437

Cipher Suites for SSL Proxy

IN THIS SECTION

Read this topic to understand more about cipher suites supports and managing digital certificates for SSL proxy on SRX Series Firewalls.

Cipher Suites

IN THIS SECTION

This topic includes the following sections:

Supported Cipher Suites

SSL proxy acts as an intermediary, performing SSL encryption and decryption between the client and the server, but neither the server nor the client can detect its presence. SSL relies on digital certificates and private-public key exchange pairs for client and server authentication to ensure secure communication.

Lets get familiar with all the terms we are going to refer in this section.

- **Digital Certificate or CA Certificate** —A digital certificate is an electronic means for verifying your identity through a trusted third party, known as a certificate authority (CA). Alternatively, you can use a self-signed certificate to attest to your identity. Each certificate contains a cryptographic key to encrypt plaintext or decrypt cyphertext.

- **Certificate Contents**—A digital certificate associates a public key with the identity of an individual entity to whom it is issuing the digital certificate. A digital certificate includes the following identification attributes:

  - Identification and signature of the Certificate Authority that issued the certificate.

  - Validity period

  - Serial number

  - Certificate issuer details

  - Information about the subject includes identifying information (the distinguished name) and the public key.

- **Cipher Suite**—A cipher suite is a set of cryptographic algorithms. An SSL cipher comprises encryption ciphers, an authentication method, and compression. On SRX Series Firewall, SSL sessions use key exchange method by which cryptographic keys are exchanged between the client and the servers using cryptographic algorithm. The kind of key exchange algorithm and the cipher suites used must be supported by both sides.

  SSL sessions use the algorithms from a cipher suite to:

  - Securely establish a secret key between two communicating parties

  - Protect the confidentiality of data in transit

Table 27 on page 378 provides the details of RSA keys supported on various SRX Series Firewalls.

Table 27: Maximum Key Sizes Supported on SRX Series Devices

SRX Series Devices	Supported RSA Key Size
SRX300, SRX320, SRX340, SRX345, SRX550, SRX1500, SRX4100, SRX4200, SRX4600, SRX5400, SRX5600, SRX5800	512 bits, 1024 bits, 2048 bits, 4096 bits

- Starting with Junos OS Release 15.1X49-D30 and Junos OS Release 17.3R1, server certificates of key size 4096 bits are supported. Prior to Junos OS Release 15.1X49-D30, server certificates with key size greater than 2048 bits were not supported because of cryptography hardware limitations.

- Starting in Junos OS Release 18.1R1, SSL proxy support is available on SRX300 and SRX320 devices. On SRX300 and SRX320 devices, server certificates with key size 4096 bits are not supported.

Table 28 on page 378 displays a list of supported ciphers. NULL ciphers are excluded.

Table 28: Supported SSL Cipher List

SSL Cipher	Key Exchange Algorithm	Data Encryption	Message Integrity	Preferred Ciphers Category	Earliest Supported Release
ECDHE-ECDSA-AES-256-GCM-SHA384	ECDHE/DSA key exchange	256-bit AES/GCM	SHA384 hash	Strong	Junos OS Release 18.3R1
ECDHE-ECDSA-AES-128-GCM-SHA256	ECDHE/DSA key exchange	128-bit AES/GCM	SHA256 hash	Strong	Junos OS Release 18.3R1
ECDHE-ECDSA-AES-256-CBC-SHA384	ECDHE/DSA key exchange	256-bit AES/CBC	SHA384 hash	Strong	Junos OS Release 18.3R1
ECDHE-ECDSA-AES-128-CBC-SHA256	ECDHE/DSA key exchange	128-bit AES/CBC	SHA256 hash	Strong	Junos OS Release 18.3R1

Table 28: Supported SSL Cipher List *(Continued)*

SSL Cipher	Key Exchange Algorithm	Data Encryption	Message Integrity	Preferred Ciphers Category	Earliest Supported Release
ECDHE-ECDSA-AES-256-CBC-SHA	ECDHE/DSA key exchange	256-bit AES/CBC	SHA hash	Strong	Junos OS Release 18.3R1
ECDHE-ECDSA-AES-128-CBC-SHA	ECDHE/DSA key exchange	128-bit AES/CBC	SHA hash	Strong	Junos OS Release 18.3R1
ECDHE-RSA-AES256-GCM-SHA384	ECDHE/RSA key exchange	256-bit AES/GCM	SHA384 hash	Strong	Junos OS Release 15.1X49-D10
ECDHE-RSA-AES256-CBC-SHA384	ECDHE/RSA key exchange	256-bit AES/CBC	SHA384 hash	Strong	Junos OS Release 15.1X49-D10
ECDHE-RSA-AES256-CBC-SHA	ECDHE/RSA key exchange	256-bit AES/CBC	SHA hash	Strong	Junos OS Release 15.1X49-D10
ECDHE-RSA-AES128-GCM-SHA256	ECDHE/RSA key exchange	128-bit AES/GCM	SHA256 hash	Strong	Junos OS Release 15.1X49-D10
ECDHE-RSA-AES128-CBC-SHA256	ECDHE/RSA key exchange	128-bit AES/CBC	SHA256 hash	Strong	Junos OS Release 15.1X49-D10
ECDHE-RSA-AES128-CBC-SHA	ECDHE/RSA key exchange	128-bit AES/CBC	SHA hash	Strong	Junos OS Release 15.1X49-D10

Table 28: Supported SSL Cipher List *(Continued)*

SSL Cipher	Key Exchange Algorithm	Data Encryption	Message Integrity	Preferred Ciphers Category	Earliest Supported Release
RSA-AES256-GCM-SHA384	ECDHE/RSA key exchange	256-bit AES/GCM	SHA384 hash	Strong	Junos OS Release 15.1X49-D10
RSA-AES256-CBC-SHA256	ECDHE/RSA key exchange	256-bit AES/CBC	SHA256 hash	Strong	Junos OS Release 15.1X49-D10
RSA-AES128-GCM-SHA256	ECDHE/RSA key exchange	128-bit AES/GCM	SHA256 hash	Strong	Junos OS Release 15.1X49-D10
RSA-AES128-CBC-SHA256	ECDHE/RSA key exchange	128-bit AES/CBC	SHA256 hash	Medium	Junos OS Release 15.1X49-D10
RSA-AES128-CBC-SHA	RSA key exchange	128-bit AES/CBC	SHA hash	Weak	Junos OS Release 12.1
RSA-AES256-CBC-SHA	RSA key exchange	256-bit AES/CBC	SHA hash	Weak	Junos OS Release 12.1

Starting in Junos OS Release 21.2R1, on SRX Series Firewalls, SSL proxy supports TLS version 1.3 and it provides improved security and better performance. Table 29 on page 380 displays a list of TLS 1.3 supported ciphers.

Table 29: TLS 1.3 Supported Cipher List

TLS Cipher	Key Exchange Algorithm	Data Encryption	Message Integrity	Earliest Supported Release
TLS_AES_256_GCM_SHA384	Any	256-bit AES/GCM	SHA384 hash	Junos OS Release 21.2R1

Table 29: TLS 1.3 Supported Cipher List *(Continued)*

TLS Cipher	Key Exchange Algorithm	Data Encryption	Message Integrity	Earliest Supported Release
TLS_AES_128_GCM_SHA256	Any	128-bit AES/GCM	SHA256 hash	Junos OS Release 21.2R1
TLS_CHACHA20_POLY1305_SHA256	Any	256-bit CHACHA20_POLY1305	SHA256 hash	Junos OS Release 21.2R1
TLS_AES_128_CCM_SHA256	Any	128-bit AES/CCM	SHA256 hash	Junos OS Release 21.2R1
TLS_AES_128_CCM_8_SHA256	Any	128-bit AES/CCM	SHA256 hash	Junos OS Release 21.2R1

Starting in Junos OS Release 18.4R1, support for some ciphers in custom ciphers are deprecated. Table 30 on page 381 provides the list of the deprecated ciphers.

Table 30: List of Deprecated Ciphers

SSL Cipher	Key Exchange Algorithm	Data Encryption	Message Integrity	Preferred Ciphers Category	Earliest Supported Release
ECDHE-ECDSA-3DES-EDE-CBC-SHA	ECDHE/DSA key exchange	3DES EDE/CBC	SHA hash	Strong	Junos OS Release 18.3R1
ECDHE-RSA-DES-CBC3-SHA	ECDHE/RSA key exchange	DES CBC	SHA hash	Medium	Junos OS Release 15.1X49-D10
RSA-RC4-128-MD5	RSA key exchange	128-bit RC4	Message Digest 5 (MD5) hash	Medium	Junos OS Release 12.1

Table 30: List of Deprecated Ciphers *(Continued)*

SSL Cipher	Key Exchange Algorithm	Data Encryption	Message Integrity	Preferred Ciphers Category	Earliest Supported Release
RSA-RC4-128-SHA	RSA key exchange	128-bit RC4	Secure Hash Algorithm (SHA) hash	Medium	Junos OS Release 12.1
RSA-EXPORT-1024-RC4-56-MD5	RSA 1024 bit export	56-bit RC4	MD5 hash	Weak	Junos OS Release 12.1
RSA-EXPORT-1024-RC4-56-SHA	RSA 1024 bit export	56-bit RC4	SHA hash	Weak	Junos OS Release 12.1
RSA-EXPORT-RC4-40-MD5	RSA-export	40-bit RC4	MD5 hash	Weak	Junos OS Release 12.1
RSA-EXPORT-DES40-CBC-SHA	RSA-export	40-bit DES/CBC	SHA hash	Weak	Junos OS Release 12.1
RSA-EXPORT-1024-DES-CBC-SHA	RSA 1024 bit export	DES/CBC	SHA hash	Weak	Junos OS Release 12.1
RSA-3DES-EDE-CBC-SHA	RSA key exchange	3DES EDE/CBC	SHA hash	Weak	Junos OS Release 12.1
RSA-DES-CBC-SHA	RSA key exchange	DES CBC	SHA hash	Weak	Junos OS Release 12.1

Note the following:

- Supported SSL ciphers for HTTPS firewall authentication are RSA-3DES-EDE-CBC-SHA , RSA-AES-128-CBC-SHA, and RSA-AES-256-CBC-SHA.
- Cipher suites that have “export” in the title are intended for use outside of the United States and might have encryption algorithms with limited key sizes. Export ciphers are not enabled by default. You need to either configure the export ciphers to enable or install a domestic package.
- ECDHE-based cipher suits support the perfect forward secrecy feature in SSL proxy.

Perfect forward secrecy is a specific key agreement protocols which ensures that all transactions sent over the Internet are secure. Perfect forward secrecy generates a unique session key for every session initiated by user. This ensures that the compromise of a single session key has no impact on data other than that exchanged in the specific session protected by that particular key.

Configuring Cipher Suites for SSL Proxy

You can use following options in SSL proxy profile configuration to set cipher suites:

- Preferred Ciphers—Preferred ciphers allow you to define an SSL cipher with acceptable key strength: strong, medium, or weak.

  If you do not want to use one of the three categories, you can select ciphers from each of the categories to form a custom cipher set. Custom ciphers allow you to define your own cipher list. To configure custom ciphers, you must set preferred-ciphers to custom. Example:

  ```
  set services ssl proxy profile profile-name preferred-ciphers custom
  ```

- Custom Ciphers—Custom ciphers allow you to define your own cipher list. Example:

  ```
  set services ssl proxy profile profile-name custom-ciphers ecdhe-ecdsa-with-aes-256-cbc-sha384
  set services ssl proxy profile profile-name custom-ciphers ecdhe-ecdsa-with-aes-128-cbc-sha256
  ```

  Starting in Junos OS Release 21.2R1, you can also use the following custom ciphers:

  ```
  set services ssl proxy profile profile-name custom-ciphers tls13-with-aes-256-gcm-sha384
  set services ssl proxy profile profile-name custom-ciphers tls13-with-aes-128-gcm-sha256
  set services ssl proxy profile profile-name custom-ciphers tls13-with-chacha20-poly1305-sha256
  set services ssl proxy profile profile-name custom-ciphers tls13-with-aes-128-ccm-sha256
  set services ssl proxy profile profile-name custom-ciphers tls13-with-aes-128-ccm8-sha256
  ```

Use the following steps to configure an SSL proxy with custom ciphers:

- Generate a root CA certificate or you can import your own trusted CA certificate and private and public keys into the device.

- Create an SSL proxy profile and associate root CA certificate (Root CA or the server certificate).

- Enable preferred-cipher in the SSL proxy as a custom-cipher and attach custom cipher

Example:

This example shows how to create a custom cipher. In this example, you set `preferred-cipher` to custom and add the cipher list (ecdhe-ecdsa-with-aes-256-cbc-sha384 and ecdhe-ecdsa-with-aes-128-cbc-sha256):

```
request security pki local-certificate load filename rootCA.pem key rootCA.key certificate-id
rootCAEcds
```

```
set services ssl proxy profile profile-name server-certificate rootCAEcds
```

Or

```
set services ssl proxy profile profile-name root-ca rootCAEcds
```

```
set services ssl proxy profile profile-name preferred-ciphers custom
```

```
set services ssl proxy profile profile-name custom-ciphers ecdhe-ecdsa-with-aes-256-cbc-sha384
set services ssl proxy profile profile-name custom-ciphers ecdhe-ecdsa-with-aes-128-cbc-sha256
```

Proceed with configuring the SSL proxy profile and applying the SSL proxy profile to a security policy

ECDSA Cipher Suite Support for SSL Proxy

Starting in Junos OS Release 18.3R1, SRX Series Firewalls support ECDSA cipher suites for SSL proxy. ECDSA is a version of the Digital Signature Algorithm (DSA) and is based on Elliptic-curve cryptography (ECC). To use ECDSA ciphers on your security device, you must ensure to:

- Include the certificates containing ECC-capable public keys on the device. Support is available for the Elliptic Curve Cryptography (ECC) certificate only with the Elliptic Prime Curve 256 bit (P-256).

- Include the ECDSA certificate option for the root CA. You can include one RSA certificate and one ECDSA certificate each. Having both ECC and RSA certificate allows you to perform ECC-based key exchange or RSA-based key exchange depending on the client and the server device’s compatibility.

- For reverse proxy, include the ECDSA certificate for the server certificate. No restriction on the number of ECDSA or RSA certificate inclusion.

- A trusted CA certificate can either be an RSA-based certificate and an ECDSA-based certificate. All features supported on an RSA-based certificate such as certificate cache, certificate revocation list (CRL), certificate chain are supported on an ECDSA certificate.

Configuring Server Certificates of Key Size 4096 Bits on SRX300 and SRX320

Starting in Junos OS Release 19.4R1, SRX300 and SRX320 devices support RSA certificates with key size 4096 bits. This support is available only when the SRX300 and SRX320 devices are operating in standalone mode.

You must explicitly configure the SSL proxy profile on SRX300 and SRX320 devices to use the server certificate with key size 4096 bits. Example:

SSL Forward Proxy Profile

```
proxy {
    profile sslfp-proxy-profile {
        trusted-ca all;
        root-ca ssl-inspect-ca;
        actions {
            allow-strong-certificate;
        }
    }
}
```

SSL Reverse Proxy Profile

```
proxy {
    profile server-protection-profile {
        server-certificate ssl-server-protection;
        actions {
            allow-strong-certificate;
        }
    }
}
```

Release History Table

Release	Description
19.4R1	Starting in Junos OS Release 19.4R1, SRX300 and SRX320 devices support RSA certificates with key size 4096 bits

18.4R1	Starting in Junos OS Release 18.4R1, support for some ciphers in custom ciphers are deprecated.
18.3R1	Starting in Junos OS Release 18.3R1, SRX Series Firewalls support ECDSA cipher suites for SSL proxy. ECDSA is a version of the Digital Signature Algorithm (DSA) and is based on Elliptic-curve cryptography (ECC).

RELATED DOCUMENTATION

SSL Proxy | 350

SSL Certificates | 355

Configuring SSL Proxy | 386

SSL Proxy Logs | 445

Operational Commands to Troubleshoot SSL Sessions | 450

Configuring SSL Proxy

IN THIS SECTION

SRX Series Firewall support SSL forward proxy and SSL reverse proxy.

Configuring SSL Forward Proxy

IN THIS SECTION

SSL Proxy Configuration Overview

"Configuring SSL Forward Proxy" on page 387 displays an overview of how SSL proxy is configured. Configuring SSL proxy includes:

- Configuring the root CA certificate
- Loading a CA profile group
- Configure SSL proxy profile and associate root CA certificate and CA profile group
- Create a security policy by defining input traffic match criteria
- Applying an SSL proxy profile to a security policy
- Optional steps such as creating allowlists and SSL proxy logging

Configuring a Root CA Certificate

A CA can issue multiple certificates in the form of a tree structure. A root certificate is the topmost certificate of the tree, the private key of which is used to *sign* other certificates. All certificates immediately below the root certificate inherit the signature or trustworthiness of the root certificate. This is somewhat like the *notarizing* of an identity.

You can configure a root CA certificate by first obtaining a root CA certificate (by either generating a self-signed one or importing one) and then applying it to an SSL proxy profile. You can obtain a root CA certificate by using the Junos OS CLI

Generate a Root CA Certificate with CLI

To define a self-signed certificate in CLI, you must provide the following details:

- Certificate identifier (generated in the previous step)
- Fully qualified domain name (FQDN) for the certificate
- e-mail address of the entity owning the certificate
- Common name and the organization involved

Generate a root CA certificate using the Junos OS CLI:

1. From operational mode, generate a PKI public/private key pair for a local digital certificate.

   ```
   user@host>request security pki generate-key-pair certificate-id certificate-id size size type
   type
   ```

   Here, you can select the one of the following combinations:

   - 1024 bits (RSA/DSA only)
   - 2048 bits (RSA/DSA only)
   - 256 bits (ECDSA only)
   - 384 bits (ECDSA only)
   - 4096 bits (RSA/DSA only)
   - 521 bits (ECDSA only)

   Example:

   ```
   user@host>request security pki generate-key-pair certificate-id SECURITY-cert size 2048 type
   rsa
   ```

Or

```
user@host>request security pki generate-key-pair certificate-id SECURITY-cert size 521 type
ecdsa
```

2. Define a self-signed certificate.

   ```
   user@host>request security pki local-certificate generate-self-signed certificate-id
   certificate-id domain-name domain-name subject subject email email-id add-ca-constraint
   ```

   Example:

   ```
   user@host> request security pki local-certificate generate-self-signed certificate-id
   SECURITY-cert domain-name labs.abc.net subject
   DC=mydomain.net,L=Sunnyvale,O=Mydomain,OU=LAB,CN=SECURITY email lab@labs.abc.net add-ca-
   constraint
   ```

   By configuring the `add-ca-constraint` option, you make sure that the certificate can be used for signing other certificates.

3. From configuration mode, apply the loaded certificate as root-ca in the SSL proxy profile.

   ```
   [edit]
   user@host# set services ssl proxy profile profile-name root-ca certificate-id
   ```

   Example:

   ```
   [edit]
   user@host# set services ssl proxy profile SECURITY-SSL-PROXY root-ca SECURITY-cert
   ```

4. Import the root CA as a trusted CA into client browsers. This is required for the client browsers to trust the certificates signed by the SRX Series Firewall. See "Importing a Root CA Certificate into a Browser" on page 392.

Configuring a CA Profile Group

The CA profile defines the certificate information for authentication. It includes the public key that SSL proxy uses when generating a new certificate. Junos OS allows you to create a group of CA profiles and load multiple certificates in one action, view information about all certificates in a group, and delete unwanted CA groups.

You can load a group of CA profiles by obtaining a list of trusted CA certificates, defining a CA group, and attaching the CA group to the SSL proxy profile.

1. Obtain a list of trusted CA certificates by using one of the following methods. When a connection is initiated, the connecting device (such as a Web browser) checks whether the certificate is issued by a trusted CA. Without these certificates, browsers cannot validate the identity of most websites and mark them as untrusted sites.

    - Junos OS provides a default list of trusted CA certificates that you can load on your system. The Junos OS package contains the default CA certificates as a PEM file (for example, **trusted_CA.pem**). After you download the Junos OS package, the default certificates are available on your system.

      From operational mode, load the default trusted CA certificates (the group name identifies the CA profile group):

      ```
      user@host> request security pki ca-certificate ca-profile-group load ca-group-name group-
      name filename default
      ```

      Example:

      ```
      user@host> request security pki ca-certificate ca-profile-group load ca-group-name
      SECURITY-CA-GROUP filename default
      ```

      The SRX Series Firewall support dynamic update of default trusted CA certificates. The latest default trusted CA certificates are stored and periodically updated on Juniper CDN server (http://signatures.juniper.net/cacert). Automatic download of trusted CA bundle is enabled by default in Junos OS device. This eliminates the need for managing these certificates manually in the event of compromise. To use this feature, see Dynamic Updated of Trusted CA Certificates.

      We recommend using this method.

    - Alternatively, you can define your own list of trusted CA certificates and import them on your system. You get the list of trusted CAs in a single PEM file (for example **IE-all.pem**) and save the PEM file in a specific location (for example, **/var/tmp**).

      From operational mode, load the trusted list to the device (the group name identifies the CA profile group):

      ```
      user@host> request security pki ca-certificate ca-profile-group load ca-group-name group-
      name filename /var/tmp/IE-all.pem
      ```

Example:

```
user@host> request security pki ca-certificate ca-profile-group load ca-group-name
SECURITY-CA-GROUP filename /var/tmp/custom-file.pem
```

- Download the latest CA bundle list from another 3rd party such as Mozilla (https://curl.haxx.se/docs/caextract.html). The list of trusted Certificate Authority can change over time, ensure that you use the latest CA bundle.

- Import your own trusted CA certificates using the Public Key Infrastructure (PKI). The PKI helps verify and authenticate the validity of the trusted CA certificates. You create CA profile groups that include trusted CA certificates, then import the group on your device for server authentication.

2. Attach the trusted CA or trusted CA group to the SSL proxy profile. You can attach all trusted CA or one trusted CA at a time:

   - Attach all CA profile groups:

   ```
   [edit]
   user@host# set services ssl proxy profile profile-name trusted-ca all
   ```

   Example

   ```
   [edit]
   user@host# set services ssl proxy profile SECURITY-SSL-PROXY trusted-ca all
   ```

   - Attach one CA profile group (the group name identifies the CA profile group).

   ```
   [edit]
   user@host# set services ssl proxy profile profile-name trusted-ca ca-name
   ```

   Example

   ```
   [edit]
   user@host# set services ssl proxy profile SECURITY-SSL-PROXY trusted-ca orgA-ca-profile
   ```

You can easily display information about all certificates in a CA profile group:

```
user@host> show security pki ca-certificates ca-profile-group group-name
```

You can delete a CA profile group. Remember that deleting a CA profile group deletes all certificates that belong to that group:

```
user@host> clear security pki ca-certificates ca-profile-group group-name
```

Importing a Root CA Certificate into a Browser

In order to have your browser or system automatically trust all certificates signed by the root CA configured in the SSL proxy profile, you must instruct your platform or browser to trust the CA root certificate.

To import a root CA certificate:

1. Generate a PEM format file for the configured root CA.

   ```
   request security pki local-certificate export certificate-id root-ca type pem filename path/file-name.pem
   ```

2. Import a root CA certificate into a browser.

   From Internet Explorer (version 8.0):

   a. From the Tools menu, select **Internet Options**.

   b. On the Content tab, click **Certificates.**

   c. Select the **Trusted Root Certification Authorities** tab and click **Import.**

   d. In the Certificate Import Wizard, navigate to the required root CA certificate and select it.

   From Firefox (version 39.0):

   a. From the Tools menu, select **Options**.

   b. From the Advanced menu, select the **Certificates** tab and click **View Certificate**.

   c. In the Certificate Manager window, select the **Authorities** tab and click **Import**.

   d. Navigate to the required root CA certificate and select it.

   From Google Chrome (45.0):

   a. From the Settings menu, select **Show Advanced Settings**.

b. From the Advanced menu, select the **Certificates** tab and click **View Certificate**.

c. Under HTTPS/SSL, click **Manage Certificates**.

d. In the Certificate window, select **Trusted Root Certification Authorities** and click **Import**.

e. In the Certificate Import Wizard, navigate to the required root CA certificate and select it.

Applying an SSL Proxy Profile to a Security Policy

SSL proxy is enabled as an application service within a security policy. In a security policy, you specify the traffic that you want the SSL proxy enabled on as match criteria and then specify the SSL proxy CA profile to be applied to the traffic. Figure 17 on page 393 displays a graphical view of SSL proxy profile and security policy configuration.

Figure 17: Applying an SSL Proxy Profile to a Security Policy

To enable SSL proxy in a security policy:

This example assumes that you have already creates security zones trust and untrust and creating a security policy for the traffic from trust zone to untrust zone.

1. Create a security policy and specify the match criteria for the policy. As match criteria, specify the traffic for which you want to enable SSL proxy.

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy policy-name match
source-address source-address
user@host# set security policies from-zone trust to-zone untrust policy policy-name match
destination-address destination-address
user@host# set security policies from-zone trust to-zone untrust policy policy-name match
application application
```

   Example:

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

2. Apply the SSL proxy profile to the security policy.

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy policy
SECURITY_POLICY then permit application-services ssl-proxy profile-name SECURITY-SSL-PROXY
```

Configuring SSL Proxy Logging

When configuring SSL proxy, you can choose to set the option to receive some or all of the logs. SSL proxy logs contain the logical system name, SSL proxy allowlists, policy information, SSL proxy information, and other information that helps you troubleshoot when there is an error.

You can configure logging of *all* or specific events, such as error, warning, and information events. You can also configure logging of sessions that are allowlisted, dropped, ignored, or allowed after an error occurs.

```
[edit]
user@host# set services ssl proxy profile profile-name actions log all
user@host# set services ssl proxy profile profile-name actions log sessions-whitelisted
user@host# set services ssl proxy profile profile-name actions log sessions-allowed
user@host# set services ssl proxy profile profile-name actions log errors
```

You can use **enable-flow-tracing** option to enable debug tracing.

Configuring Certificate Authority Profiles

A certificate authority (CA) profile configuration contains information specific to a CA. You can have multiple CA profiles on an SRX Series Firewall. For example, you might have one profile for orgA and one for orgB. Each profile is associated with a CA certificate. If you want to load a new CA certificate without removing the older one then create a new CA profile (for example, Microsoft-2008). You can group multiple CA profiles in one trusted CA group for a given topology.

In this example, you create a CA profile called ca-profile-security with CA identity microsoft-2008. You then create proxy profile to the CA profile.

1. From configuration mode, configure the CA profile used for loading the certificate.

```
[edit]
user@host# set security pki ca-profile profile-name ca-identity ca-identity
```

Example:

```
user@host# set security pki ca-profile ca-profile-security ca-identity example.com
```

2. Commit the configuration.

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

3. From operational mode, load the certificate using PKI commands.

```
user@host> request security pki ca-certificate load ca-profile profile-name filename filename
```

Example:

```
user@host> request security pki ca-certificate load ca-profile ca-profile-security filename
srx-123.crt
```

4. From configuration mode, disable the revocation check (if required).

```
[edit]
user@host# set security pki ca-profile profile-name ca-identity ca-identity revocation-check
disable
```

Example:

```
[edit]
user@host# set security pki ca-profile ca-profile-security ca-identity example.com revocation-
check disable
```

5. From configuration mode, configure the loaded certificate as a trusted CA in the SSL proxy profile.

```
[edit]
user@host# set services ssl proxy profile ssl-proxy-profile-name trusted-ca ca-profile-name
```

Example:

```
[edit]
user@host# set services ssl proxy profile ssl-proxy-sample trusted-ca ca-profile-security
```

NOTE: More than one trusted CA can be configured for a profile.

6. (Optional) If you have multiple trusted CA certificates, you do not have to specify each trusted CA separately. You can load *all* the trusted CA certificates using the following command from configuration mode.

```
[edit]
user@host# set services ssl proxy profile ssl-proxy-profile-name root-ca ssl-inspect-ca
user@host# set services ssl proxy profile ssl-proxy-profile-name trusted-ca all
```

NOTE: Alternatively, you can import a set of trusted CAs from your browser into the SRX Series Firewall.

Exporting Certificates to a Specified Location

When a self-signed certificate is generated using a PKI command, the newly generated certificate is stored in a predefined location (**var/db/certs/common/local**).

Use the following command to export the certificate to a specific location (within the device). You can specify the certificate ID, the filename, and the type of file format (DER/PEM):

```
user@host> request security pki local-certificate export certificate-id certificate-id filename
filename  type der
```

Ignoring Server Authentication

Junos OS allows you to configure an option to ignore server authentication completely. If you configure your system to ignore authentication, then any errors encountered during server certificate verification at the time of the SSL handshake are ignored. Commonly ignored errors include the inability to verify CA signature, incorrect certificate expiration dates, and so forth. If this option is not set, all the sessions where the server sends self-signed certificates are dropped when errors are encountered.

We do not recommend using this option for authentication because configuring it results in websites not being authenticated at all. However, you can use this option to effectively identify the root cause of dropped SSL sessions.

From configuration mode, specify to ignore server authentication:

```
[edit]
user@host# set services ssl proxy profile profile-name actions ignore-server-auth-failure
```

SSL Reverse Proxy

IN THIS SECTION

Overview

The proxy model implementation for server protection (often called *reverse proxy*) is supported on SRX Series Firewalls to provide improved handshaking and support for more protocol versions. You can enable Layer 7 services (application security, IPS, Content Security, ATP Cloud) on the traffic decrypted by SSL reverse proxy.

Starting in Junos OS Release 15.1X49-D80 and 17.3R1, SSL reverse proxy is supported on SRX5000 line, SRX4100, SRX4200, SRX1500 devices.

Starting in Junos OS Release 15.1X49-D80 and 17.3R1, we recommend using the SSL reverse proxy and Intrusion Detection and Prevention (IDP) instead of using the IDP SSL inspection functionality.

Starting from Junos OS 15.1X49-D80 and 17.3R1, IDP SSL Inspection is deprecated—rather than immediately removed—to provide backward compatibility and a chance to bring your configuration into compliance with the new configuration.

No Link Title provides the changes applicable on SRX Series Firewalls post 15.1X48-D80 and 17.3R1 releases.

Table 31: Comparing Reverse Proxy Before and After Junos OS Release 15.1X49-D80

Feature	Prior to 15.1X49-D80	15.1X49-D80 and 17.3R1 later
Proxy model	Runs only in tap mode Instead of participating in SSL handshake, it listens to the SSL handshake, computes session keys and then decrypts the SSL traffic.	Terminates client SSL on the SRX Series Firewall and initiates a new SSL connection with a server. Decrypts SSL traffic from the client/server and encrypts again (after inspection) before sending to the server/client.
Protocol version	Does not support TLS Version 1.1 and 1.2.	Supports all current protocol versions.
Key exchange methods	• Supports RSA • Does not support DHE.	• Supports RSA • Support DHE or ECDHE
Echo system	Tightly coupled with IDP engine and its detector.	Uses existing SSL forward proxy with TCP proxy underneath.
Security services	Decrypted SSL traffic can be inspected only by IDP.	Just like forward proxy, decrypted SSL traffic is available for all security services.
Ciphers supported	Limited set of ciphers are supported.	All commonly used ciphers are supported.

You must configure either `root-ca` or `server-certificate` in an SSL proxy profile. Otherwise the commit check fails. See No Link Title.

Table 32: Supported SSL Proxy Configurations

server-certificate configured	root-ca configured	Profile type
No	No	Commit check fails. You must configure either `server-certificate` or `root-ca`.
Yes	Yes	Commit check fails. Configuring both `server-certificate` and `root-ca` in the same profile is not supported.

Table 32: Supported SSL Proxy Configurations *(Continued)*

server-certificate configured	root-ca configured	Profile type
No	Yes	Forward proxy
Yes	No	Reverse proxy

Configuring multiple instances of forward and reverse proxy profiles are supported. But for a given firewall policy, only one profile (either a forward or reverse proxy profile) can be configured. Configuring both forward and reverse proxy on the same device is also supported.

You cannot configure the previous reverse proxy implementation with the new reverse proxy implementation for a given firewall policy. If both are configured, you will receive a commit check failure message.

The following are the minimum steps to configure reverse proxy:

1. Load the server certificates and their keys into the SRX Series Firewall certificate repository using the CLI command `request security pki local-certificate load filename` *`filename`* `key key certificate-id` *`certificate-id`* `passphrase` *`exmample@1234`*. For example:

   ```
   user@host> request security pki local-certificate load filename /cf0/cert1.pem key /cf0/
   key1.pem certificate-id server1_cert_id passphrase example@1234
   ```

2. Attach the server certificate identifier to the SSL Proxy profile using the CLI command `set services ssl proxy profile` *`profile`* `server-certificate` *`certificate-id`* `passphrase` *`exmample@1234`*. For example

   user@host# set services ssl proxy profile server-protection-profile server-certificate server2_cert_id

3. Use the `show services ssl` CLI command to verify your configuration. For example:

   ```
   user@host# show services ssl
   profile server-protection-profile {

           server-certificate [server1_cert_id , server2_cert_id];
           actions {
            logs {
                 all;
                 }
   ```

```
             }
}
```

The SSL forward proxy and reverse proxy require a profile to be configured at the firewall rule level. In addition, you must also configure server certificates with private keys for reverse proxy. During an SSL handshake, the SSL proxy performs a lookup for a matching server private key in its server private key hash table database. If the lookup is successful, the handshake continues. Otherwise, SSL proxy terminates the hand shake. Reverse proxy does not prohibit server certificates. It forwards the actual server certificate/chain as is to the client without modifying it. Intercepting the server certificate occurs only with forward proxy.

The following shows example forward and reverse proxy profile configurations.

```
# show services ssl
...
proxy {
Â  Â  profile ssl-inspect-profile-dut { # For forward proxy. No server cert/key is needed.
            trusted-ca all;
Â  Â  Â  Â  root-ca ssl-inspect-ca;
Â  Â  Â  Â  actions {
Â  Â  Â  Â  Â  Â  ignore-server-auth-failure;
Â  Â  Â  Â  Â  Â  log {
Â  Â  Â  Â  Â  Â  Â  Â  all;
Â  Â  Â  Â  Â  Â  }
Â  Â  Â  Â  }
Â  Â  }
Â  Â  profile ssl-1 {
            trusted-ca all;
Â  Â  Â  Â  root-ca ssl-inspect-ca;
Â  Â  Â  Â  actions {
Â  Â  Â  Â  Â  Â  ignore-server-auth-failure;
Â          Â  Â  Â  Â  Â  log {
Â  Â  Â  Â  Â  Â  Â  Â  all;
Â  Â  Â  Â  Â  Â  }
Â  Â  Â  Â  }
Â  Â  }
    profile ssl-2 {
            trusted-ca all;
Â  Â  Â  Â  root-ca ssl-inspect-ca;
Â  Â  Â  Â  actions {
Â  Â  Â  Â  Â  Â  ignore-server-auth-failure;
Â  Â  Â  Â  Â  Â  log {
```

```
Â  Â  Â  Â  Â  Â  Â  Â  all;
Â  Â  Â  Â  Â  Â  }
Â  Â  Â  Â  }
Â  Â  }
Â  Â  profile ssl-server-protection { # For reverse proxy. No root-ca is needed.

Â  Â  Â  Â  server-certificate ssl-server-protection;
Â  Â  Â  Â  actions {
Â  Â  Â  Â  Â  Â  log {
Â  Â  Â  Â  Â  Â  Â  Â  all;
Â  Â  Â  Â  Â  Â  }
Â  Â  Â  Â  }
Â  Â  }
}
...
```

Configuring the SSL Reverse Proxy

This example shows how to configure reverse proxy to enable server protection. For server protection, additionally, server certificate(s) with private key(s) must be configured.

A reverse proxy protects servers by hiding the details of the servers from the clients, there by adding an extra layer of security.

To configure an SSL reverse proxy, you must:

- Load the server certificate(s) and their key(s) into SRX Series Firewall's certificate repository.
- Attach the server certificate identifier(s) to the SSL proxy profile.
- Apply SSL proxy profile as application services in a security policy.

To configure SSL reverse proxy:

1. Load the signing certificate and the respective key for the SSL proxy profile in PKI memory.

```
user@host>  request security pki local-certificate load filename /cf0/cert1.pem   key /cf0/
key1.pem certificate-id server1_cert_id
```

2. Attach the server certificate to the SSL proxy profile.

```
user@host# set services ssl proxy profile server-protection-profile server-certificate
server1_cert_id
```

3. Create a security policy and specify the match criteria for the policy. As match criteria, specify the traffic for which you want to enable SSL proxy.

```
user@host# set security policies from-zone untrust to-zone trust policy 1 match source-address
any
user@host# set security policies from-zone untrust to-zone trust policy 1 match destination-
address any
user@host# set security policies from-zone untrust to-zone trust policy 1 match application any
```

4. Apply the SSL proxy profile to the security policy. This example assumes that security zones are created as per requirements.

```
user@host# set security policies from-zone untrust to-zone trust policy 1 then permit
application-services ssl-proxy server-protection-profile
```

Verifying the SSL Reverse Proxy Configuration on the Device

IN THIS SECTION

Purpose

Viewing the SSL reverse proxy statistics on the SRX Series Firewall.

Action

You can view the SSL proxy statistics by using the `show services ssl proxy statistics` command.

```
root@host> show services ssl proxy statistics
        PIC:spu-1 fpc[0] pic[1] ------
                sessions matched                    0
                sessions whitelisted                0
                sessions bypassed:non-ssl           0
                sessions bypassed:mem overflow      0
                sessions bypassed:low memory        0
                sessions created                          0
                sessions ignored                          0
                sessions active                           0
                sessions dropped                          0
```

Configure SSL Proxy with Content Security

IN THIS SECTION

SRX Series Firewalls supports client protection (forward proxy) and server protection (reverse proxy). You can configure SSL proxy profile for forward proxy and reverse proxy with Content Security enabled.

Configure SSL Forward Proxy with Content Security

In this procedure, you configure an SSL forward proxy profile with Content Security. When you configure Content Security, the SSL proxy acts as an SSL server by terminating the SSL session from the client and establishing a new SSL session to the server. The SRX Series Firewall decrypts and then reencrypts all SSL proxy traffic. Content Security can use the decrypted content from SSL proxy.

Generate local certificate as root-ca.

1. From operational mode, generate a key pair for a local digital certificate.

   ```
   user@host> request security pki generate-key-pair certificate-id certificate-id size size type
   type
   ```

2. Generate local certificate using the key pair generated above.

   ```
   user@host> request security pki local-certificate generate-self-signed certificate-id
   certificate-id domain-name domain-name subject subject email email-id
   ```

3. From configuration mode, apply the loaded certificate as root-ca in the SSL proxy profile.

```
[edit]
user@host# set services ssl proxy profile profile-name root-ca value
user@host# set services ssl proxy profile profile-name actions ignore-server-auth-failure
```

4. Attach SSL profile and Content Security policy to security policy.

```
[edit security policies from-zone trust to-zone untrust]
user@host# set policy policy-name match source-address any
user@host# set policy policy-name  match destination-address any
user@host# set policy policy-name match application any
user@host# set policy policy-name then permit application-services ssl-proxy profile-name
profile-name
user@host# set policy policy-name then application-services utm-policy utm-policy
```

Configure SSL Reverse Proxy with Content Security

In this procedure, you configure an SSL reverse proxy profile with Content Security.

1. Load the server certificates and their keys into the SRX Series Firewall certificate repository.

```
user@host> request security pki local-certificate load filename /var/tmp/certs/server-cert.pem
key /var/tmp/certs/server-key.pem certificate-id certificate-id
```

2. From configuration mode, attach the server certificate identifier to the SSL Proxy profile.

```
user@host# set services ssl proxy profile profile-name server-certificate server-cert-id
```

3. Attach SSL profile and Content Security policy to security policy for the traffic from an untrust zone to the trust zone.

```
[edit security policies from-zone untrust to-zone trust]
user@host# set policy policy-name match source-address any
user@host# set policy policy-name match destination-address server-ip-address
user@host# set policy policy-name match application any
user@host# set policy policy-name then permit application-services ssl-proxy profile-name
profile-name
user@host# set policy policy-name then application-services utm-policy utm-policy
```

RELATED DOCUMENTATION

SSL Proxy Overview

Creating an Allowlist of Exempted Destinations for SSL Proxy

SSL encryption and decryption might consume memory resources on the SRX Series Firewalls. To limit this, you can selectively bypass SSL proxy processing for some sessions such as sessions that transacts with familiar trusted servers or domains. You can also exempt the sessions with financial and banking sites due to legal requirements.

To exempt the sessions from SSL proxy, you can create an allowlist by adding IP addresses or domain names of the servers. Allowlists include addresses that you want to exempt from undergoing SSL proxy processing.

Use the following steps to create allowlist:

- Specify IP addresses and domain name in your global address book.
- Refer the global address book in SSL proxy profile.

You can configure the following types of the IP addresses in global address book.

- IPv4 addresses (plain text). For example:

```
set security address-book global address address-4 192.0.2.117
```

- IPv4 address range. For example:

```
set security address-book global address address-2 range-address 192.0.2.117 to 192.0.2.199
```

- IPv4 wildcard. For example:

```
set security address-book global address address-3 wildcard-address 203.0.113.0/24
```

- DNS name. For example:

```
set security address-book global address address-1 dns-name www.abc.com
```

- IPv6 address. For example:

  ```
  set security address-book global address address-5 FE80::/10
  ```

Allowlists do not support the following types of IP addresses:

- Translated IP addresses. Sessions are allowlisted based on the actual IP address and not on the translated IP address. Because of this, in the allowlist configuration of the SSL proxy profile, the actual IP address should be provided and not the translated IP address.
- Noncontiguous netmasks. For example:
  - IP address -203.0.113.0 and mask 255.255.255.0 that is 203.0.113.0/24 is supported.
  - IP address - 203.0.113.9 and mask 255.0.255.0 is not supported.

Following example shows you how to use allowlists in SSL proxy profile.

In this example, you exempt all sessions to `www.mycompany.com`. For this, you first specify the domain in the address book and then configure the address in the SSL proxy profile.

1. Configure the domain in the address book.

   ```
   [edit]
   user@host# set security address-book global address address-1 dns-name www.mycompany.com
   ```

2. Specify the global address book address in the SSL proxy profile.

   ```
   [edit]
   user@host# set services ssl proxy profile profile-1 whitelist address-1
   ```

Creating an Allowlist of Exempted URL Categories for SSL Proxy

IN THIS SECTION

You can configure the URL categories supported in Content Security module to exempt from SSL inspection on SRX Series Firewall. To use URL categories from Content Security, SRX Series Firewall integrates the SSL proxy profile with the EWF feature. With this now, you can configure a list of URL categories under an SSL proxy profile as allowlist along with address-books. You can configure the list from the predefined set of URL categories or custom URL categories supported by Content Security.

The security device uses the Server Name Indication (SNI) field extracted by the Content Security module to determine the URL category. The SSL proxy uses this information to determines whether to accept, and proxy, or to ignore the session.

This feature is supported on all SRX Series Firewalls and vSRX Virtual Firewalls

Starting with Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, SSL proxy allowlisting feature includes URL categories supported by Content Security.

Starting with Junos OS Release 17.4R1, SSL proxy allowlisting feature extends support to custom URL categories supported by Content Security.

Following examples show how to configure the URL categories in SSL proxy profile:

Creating an Allowlist of Exempted URL Categories

Use the following steps to configure the predefined URL categories in an SSL proxy profile.

1. The predefined URL categories depends on Content Security. To enable URL-based allowlisting in SSL proxy, the following basic URL configurations are required:

```
[edit]
user@host# set security utm utm-policy UTM-POLICY-1 web-filtering http-profile junos-wf-enhanced-default
```

2. Specify the predefined URL category in SSL proxy profile. In this example, you are using the URL category Enhanced_Financial_Data_and_Services.

```
[edit]
user@host# set services ssl proxy profile pr1 whitelist-url-categories
Enhanced_Financial_Data_and_Services
```

3. Create the security policy by specifying the match conditions and attach the Content Security policy to the security policy to use URL categories in SSL allowlist.

```
user@host# set security policies from-zone trust to-zone untrust policy p1 match source-address any
user@host# set security policies from-zone trust to-zone untrust policy p1 match destination-
```

```
address any
user@host# set security policies from-zone trust to-zone untrust policy p1 match application
any
user@host# set security policies from-zone trust to-zone untrust policy p1 permit application-
services utm-policy UTM-POLICY-1
user@host# set security policies from-zone trust to-zone untrust policy p1 permit application-
services ssl-proxy profile-name pr1
```

Creating an Allowlist of Exempted Custom URL Categories

Use the following steps to configure custom URL categories in an SSL proxy profile.

1. Create a custom URL category.

```
[edit]
user@host# set security utm custom-objects url-pattern URL-1 value www.example.com
user@host# set security utm custom-objects custom-url-category CATEGORY-1 value URL-1
user@host#  set security utm feature-profile web-filtering juniper-local profile PROFILE-1
category CATEGORY-1 action permit
```

2. Configure a Content Security policy for the Web-filtering HTTP protocol and associate the profile you created in previous step to the the Content Security policy.

```
[edit]
user@host#  set security utm utm-policy UTM-POLICY-1 web-filtering http-profile PROFILE-1
```

3. Specify the custom URL category you created in previous step in SSL proxy profile.

```
user@host# set services ssl proxy profile pr1 whitelist-url-categories CATEGORY-1
```

4. Create a security policy by specifying the match conditions and attach the Content Security policy to the security policy to use URL categories in SSL allowlist.

```
[edit]
user@host# set security policies from-zone trust to-zone untrust policy p1 match source-
address any
user@host# set security policies from-zone trust to-zone untrust policy p1 match destination-
address any
user@host# set security policies from-zone trust to-zone untrust policy p1 match application
any
```

```
user@host# set security policies from-zone trust to-zone untrust policy p1 permit application-services utm-policy UTM-POLICY-1
user@host# set security policies from-zone trust to-zone untrust policy p1 permit application-services ssl-proxy profile-name pr1
```

Release History Table

Release	Description
17.4R1	Starting with Junos OS Release 17.4R1, SSL proxy allowlisting feature extends support to custom URL categories supported by Content Security.
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and 17.3R1, SSL reverse proxy is supported on SRX5000 line, SRX4100, SRX4200, SRX1500 devices
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80 and 17.3R1, we recommend using the SSL reverse proxy and Intrusion Detection and Prevention (IDP) instead of using the IDP SSL inspection functionality.
15.1X49-D80	Starting from Junos OS 15.1X49-D80 and 17.3R1, IDP SSL Inspection is deprecated—rather than immediately removed—to provide backward compatibility and a chance to bring your configuration into compliance with the new configuration.
15.1X49-D80	Starting with Junos OS Release 15.1X49-D80 and Junos OS Release 17.3R1, SSL proxy allowlisting feature includes URL categories supported by Content Security.

RELATED DOCUMENTATION

Example: Loading CA and Local Certificates Manually

Example: Configuring a Device for Peer Certificate Chain Validation

Unified Policies for SSL Proxy

IN THIS SECTION

Application Security Services with SSL Proxy

IN THIS SECTION

With the implementation of SSL proxy, AppID can identify applications encrypted in SSL. SSL proxy can be enabled as an application service in a regular firewall policy rule. Intrusion Detection and Prevention (IDP), application firewall (AppFW), application tracking (AppTrack), advanced policy-based routing (APBR) services, Content Security, ATP Cloud, and Security Intelligence (SecIntel) can use the decrypted content from SSL proxy.

To determine if a feature is supported by a specific platform or Junos OS release, refer Feature Explorer

On the SSL payload, IDP can inspect attacks and anomalies; for example, HTTP chunk length overflow on HTTPS. On encrypted applications, such as Facebook, AppFW can enforce policies and AppTrack (when configured in the from and to zones) can report logging issues based on dynamic applications.

NOTE: If none of the services (AppFW, IDP, or AppTrack) are configured, then SSL proxy services are bypassed even if an SSL proxy is attached to a firewall policy.

NOTE: The IDP module will not perform an SSL inspection on a session if an SSL proxy is enabled for that session. That is, if both SSL inspection and SSL proxy are enabled on a session, SSL proxy will always take precedence.

Leveraging Dynamic Application Identification

SSL proxy uses application identification services to dynamically detect if a particular session is SSL encrypted. SSL proxies are allowed only if a session is SSL encrypted. The following rules apply for a session:

- Session is marked **Encrypted=Yes** in the application system cache. If the session is marked **Encrypted=Yes**, it indicates that the final match from application identification for that session is SSL encrypted, and SSL proxy transitions to a state where proxy functionality can be initiated.

- Session is marked **Encrypted=No** in the application system cache. If a non-SSL entry is found in the application system cache, it indicates that the final match from application identification for that session is non-SSL and SSL proxy ignores the session.

- An entry is not found in the application system cache. This can happen on the first session, or when the application system cache has been cleaned or has expired. In such a scenario, SSL proxy cannot wait for the final match (requires traffic in both directions). In SSL proxy, traffic in reverse direction happens only if SSL proxy has initiated an SSL handshake. Initially, for such a scenario SSL proxy tries to leverage prematch or aggressive match results from application identification , and if the results indicate SSL, SSL proxy will go ahead with the handshake.

- Application identification fails due to resource constraints and other errors. Whenever the result from application identification is not available, SSL proxy will assume static port binding and will try to initiate SSL handshake on the session. This will succeed for actual SSL sessions, but it will result in dropped sessions for non SSL sessions.

SEE ALSO

Example: Configuring Application Firewall When SSL Proxy Is Enabled | **161**
Example: Configuring Application Tracking When SSL Proxy Is Enabled | **185**

SSL Proxy Support for Unified Policies

IN THIS SECTION

Starting from Junos OS Release 18.2R1, unified policies are supported on SRX Series Firewalls, allowing granular control and enforcement of dynamic Layer 7 applications, within the traditional security policy.

Unified policies are the security policies that enable you to use dynamic applications as match conditions as part of the existing 5-tuple or 6-tuple (5-tuple with user firewall) match conditions to detect application changes over time.

SSL proxy functionality is supported when the device is configured with unified policies. As a part of this enhancement, you can configure a default SSL proxy profile.

During the initial policy lookup phase, which occurs prior to a dynamic application being identified, if there are multiple policies present in the potential policy list which contains different SSL proxy profiles, the SRX Series Firewall applies the default SSL proxy profile until a more explicit match has occurred.

We recommend that you create a default SSL proxy profile. The sessions are dropped in case of policy conflicts, if there is no default SSL proxy profile available.

You can configure an SSL proxy profile under the `[edit services ssl proxy]` hierarchy level, and then apply it as a default SSL proxy profile under the `[edit security ngfw]` hierarchy level. This configuration does not impact the existing SSL service configuration.

Configuring a default SSL proxy profile is supported for both SSL forward and reverse proxy.

Understanding How SSL Proxy Default Profile Works

Table 33 on page 412 summarizes the default SSL proxy profile behavior in unified policies.

Table 33: SSL Proxy Profile Usage in Unified Policies

Application Identification Status	SSL Proxy Profile Usage	Action
No security policy conflict	SSL proxy profile is applied when traffic matches the security policy.	SSL proxy profile is applied.
Security policy conflict (conflicting polices have distinct SSL proxy profiles)	Default SSL proxy profile is not configured or not found.	Session is terminated, because the default SSL proxy profile is not configured.
	Default SSL proxy profile is configured.	Default SSL proxy profile is applied.

Table 33: SSL Proxy Profile Usage in Unified Policies *(Continued)*

Application Identification Status	SSL Proxy Profile Usage	Action
Final application is identified	Matching security policy has a SSL proxy profile that is same as default SSL proxy profile.	Default SSL proxy profile is applied.
	Matching security policy does not have a SSL proxy profile.	Default SSL proxy profile is applied.
	Matching security policy has a SSL proxy profile that is different from the default SSL proxy profile that is already applied.	Default SSL proxy profile that is already applied, continues remain as applied.

NOTE: A security policy can have either an SSL reverse proxy profile or an SSL forward proxy profile configured at a time.

If a security policy has an SSL forward proxy profile and another security policy has an SSL reverse proxy profile, in such case, a default profile—either from SSL reverse proxy profile or from SSL forward proxy profile is considered.

⚠ **CAUTION**: We recommend creating default SSL proxy profile because sessions are dropped in case of policy conflicts, when there is no default SSL proxy profile available. A system log message is generated to log the event.

TIP: Example of the system log message:

```
"<14>1 2018-03-07T03:18:33.374-08:00 4.0.0.254 kurinji junos-ssl-proxy -
SSL_PROXY_SSL_SESSION_DROP [junos@2636.1.1.1.2.105 logical-system-name="root-logical-system"
session-id="15" source-address="4.0.0.1" source-port="37010" destination-address="5.0.0.1"
destination-port="443" nat-source-address="4.0.0.1" nat-source-port="37010" nat-destination-
address="5.0.0.1" nat-destination-port="443" profile-name="(null)" source-zone-name="untrust"
source-interface-name="xe-2/2/1.0" destination-zone-name="trust" destination-interface-
name="xe-2/2/2.0" message="default ssl-proxy profile is not configured"]
```

Default SSL Proxy Profiles in Different Scenarios

IN THIS SECTION

No Policy Conflict—All Policies Have Same SSL Proxy Profile

All matching policies have same SSL proxy profile as shown in Table 34 on page 414.

Table 34: No Policy Conflict—All Policies Have Same SSL Proxy Profile

Security Policy	Source Zone	Source IP Address	Destination Zone	Destination IP Address	Port Number	Protocol	Dynamic Application	Service	Default SSL Proxy Profile
Policy-P1	S1	Any	D1	Any	Any	Any	Facebook	SSL Proxy	SSL-1
Policy-P2	S1	Any	D1	Any	Any	Any	Google	SSL Proxy	SSL-1

In this case, both Policy-P1 and Policy-P2 have the same SSL proxy profile (SSL-1). Because there is no conflict, the profile SSL-1 is applied.

If you have configured a default SSL proxy profile (SSL-2), it is not applied. Because there is no conflict in the policies (Policy-P1 and Policy-P2).

No Policy Conflict—All Policies Have Same SSL Proxy Profile and Final Policy Has No SSL Profile

Policy-P1 and Policy-P2 have same SSL proxy profile and the Policy-3 has no SSL profile as shown in Table 35 on page 415.

Table 35: No Policy Conflict—All Policies Have Same SSL Proxy Profile and Final Policy Has No SSL Profile Configured

Security Policy	Source Zone	Source IP Address	Destination Zone	Destination IP Address	Port Number	Protocol	Dynamic Application	Service	Default SSL Proxy Profile
Policy-P1	S1	Any	D1	Any	Any	Any	Facebook	SSL Proxy	SSL-1
Policy-P3	S1	50.1.1.1	D1	Any	Any	Any	YouTube	SSL Proxy	SSL-1
Policy-P2	S1	Any	D1	Any	Any	Any	Google	Other	None

In this scenario, both Policy-P1 and Policy-P2 have the same SSL proxy profile (SSL-1). Because there is no conflict, the profile SSL-1 is applied before the final policy match.

When the final application is identified, the security policy matching with the final application, that is, Policy-P3 is applied. Because the Policy-P3 has no SSL proxy profile, the already applied profile SSL-1 remains applied. This is because, the SSL proxy profile is already applied on the traffic.

Policy Conflict—No SSL Profile Configured for Final Policy

The default SSL proxy profile is applied during potential match as shown in Table 36 on page 416. The final policy, Policy-P3 does not have any SSL proxy profile.

Table 36: Policy Conflict—No SSL Profile Configured for Final Policy

Security Policy	Source Zone	Source IP Address	Destination Zone	Destination IP Address	Port Number	Protocol	Dynamic Application	Service	Default SSL Proxy Profile
Policy-P1	S1	50.1.1.1	D1	Any	Any	Any	Facebook	SSL Proxy	SSL-1
Policy-P2	S1	50.1.1.1	D1	Any	Any	Any	Google	SSL Proxy	SSL-2
Policy-P3	S1	50.1.1.1	D1	Any	Any	Any	YouTube	Other	NA

In this example, SSL proxy profile SSL-1 is configured as default SSL proxy profile. During the policy conflict for Policy-P1 and Policy-P2, the default profile SSL-1 is applied.

When the final application is identified, the security policy matching with the final application, that is, Policy-P3 is applied. Because the Policy-P3 has no SSL proxy profile, the already applied profile SSL-1 continues to remain as applied. This is because, the SSL proxy profile is applied on the traffic.

Policy Conflict–Default SSL Proxy Profile and Different SSL Proxy Profile for Final Policy

The SSL proxy profile SSL-1 is configured as a default SSL proxy profile and is already applied before the final policy is matched. Refer Table 37 on page 416.

Table 37: Policy Conflict—Default SSL Proxy Profile and Different SSL Proxy Profile for Final Policy

Security Policy	Source Zone	Source IP Address	Destination Zone	Destination IP Address	Port Number	Protocol	Dynamic Application	Service	Default SSL Proxy Profile
Policy-P1	S1	50.1.1.1	D1	Any	Any	Any	Facebook	SSL Proxy	SSL-1

Table 37: Policy Conflict—Default SSL Proxy Profile and Different SSL Proxy Profile for Final Policy *(Continued)*

Security Policy	Source Zone	Source IP Address	Destination Zone	Destination IP Address	Port Number	Protocol	Dynamic Application	Service	Default SSL Proxy Profile
Policy-P2	S1	50.1.1.1	D1	Any	Any	Any	Google	SSL Proxy	SSL-2
Policy-P3	S1	50.1.1.1	D1	Any	Any	Any	YouTube	SSL Proxy	SSL-3

When the final application is identified, the security policy matching with the final application, that is, Policy-P3 is applied. The SSL profile for the Policy-P3, that is, SSL-3 is not applied. Instead, the SSL proxy profile SSL-2 configured and applied as default profile, continues to remain as applied.

Switching from the default SSL proxy profile that is already applied to the traffic, to another SSL proxy profile is not supported.

Limitations of SSL Proxy with Unified Policies

- When a default SSL proxy profile is enabled, it cannot be disabled even if the final security policy does not have SSL proxy configured.
- When a default SSL proxy profile is enabled and applied on the traffic and the final security policy has a different SSL proxy profile configured other than default profile, switching from the default SSL proxy profile to the SSL proxy profile in the security policy is not supported.

Configuring Default SSL Proxy Profiles

IN THIS SECTION

SSL proxy is enabled as an application service within a security policy. In a security policy, specify the match criteria for the traffic that must be SSL proxy enabled. Next, specify the SSL proxy profile to be applied to the traffic. When configuring unified policies, the steps include defining the SSL profile, then adding the SSL profile as default profile under the `[edit security ngfw]` hierarchy level, and then including to it in the desired security policy.

Configuring Default Profile for SSL Forward Proxy

In this procedure, you configure an SSL forward proxy profile, and specify the profile as the default profile.

1. Create an SSL profile and attach the CA profile group to the SSL proxy profile.

```
user@host# set services ssl proxy profile profle-name trusted-ca all
```

2. Apply the signing certificate as root-ca in the SSL proxy profile.

```
user@host# set services ssl proxy profile profle-name root-ca ssl-inspect-ca
```

3. Define the SSL proxy profile as the default profile.

```
user@host# set security ngfw default-profile ssl-proxy profle-name
```

Configuring Default Profile for SSL Reverse Proxy

In this procedure, you configure an SSL reverse proxy profile and specify the profile as the default profile.

1. Create an SSL profile and attach the CA profile group to the SSL proxy profile.

```
user@host# set services ssl proxy profile server-protection-profile server-certificate
server1_certificate-id
```

2. Define the SSL reverse proxy profile as the default profile.

```
user@host# set security ngfw default-profile ssl-proxy profile-name server-protection-profile
```

Configuring Default SSL Profiles for Logical System

In this procedure, you assign the SSL forward proxy profile or the SSL reverse proxy profile as the default profile in logical system configurations. In this case, one profile can be a default profile either from the SSL forward proxy or from the SSL reverse proxy.

- Define the SSL forward proxy profile as the default profile.

  ```
  user@host# set logical-systems LSYS1 security ngfw default-profile ssl-proxy profle-name
  ```

- Define the SSL reverse proxy profile as the default profile.

  ```
  user@host# set logical-systems LSYS1 security ngfw default-profile ssl-proxy profle-name
  ```

Example: Configuring Default SSL Proxy Profile for Unified Policy

IN THIS SECTION

This example shows how to configure a default SSL proxy profile and apply it in a unified policy.

Configuration

Step-by-Step Procedure

To configure a default SSL proxy profile and apply it in a unified policy:

1. Create an SSL profile and attach the CA profile group to the SSL proxy profile.

   ```
   user@host# set services ssl proxy profile SSL-FP-PROFILE-1 trusted-ca all
   ```

2. Apply the signing certificate as root-ca in the SSL proxy profile.

```
user@host# set services ssl proxy profile SSL-FP-PROFILE-1 root-ca ssl-inspect-ca
```

3. Define the SSL proxy profile as the default profile.

```
user@host#  set security ngfw default-profile ssl-proxy profile-name SSL-FP-PROFILE-1
```

4. Create a unified policy and specify the dynamic application as the match criteria.

```
user@host# set security policies from-zone untrust to-zone trust policy from_internet match
source-address any
user@host# set security policies from-zone untrust to-zone trust policy from_internet match
destination-address any
user@host# set security policies from-zone untrust to-zone trust policy from_internet match
application any
user@host# set security policies from-zone untrust to-zone trust policy from_internet match
dynamic-application junos:web
```

5. Apply the SSL proxy profile to the permitted traffic in the security policy.

```
user@host# set security policies from-zone untrust to-zone trust policy from_internet then
permit application-services ssl-proxy profile-name SSL-FP-PROFILE-1
```

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall with Junos OS Release 18.2R1 or later. This configuration example is tested for Junos OS Release 18.2R1.

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

Overview

In this example, you configure an SSL forward proxy profile by specifying the root CA certificate. Next, configure the profile as default SSL proxy profile. Now, you create a unified policy and invoke the SSL proxy as application services on the permitted traffic.

Verification

IN THIS SECTION

Verify SSL Proxy Configuration

Purpose

Confirm that the configuration is working properly by displaying the SSL proxy statistics.

Action

From operational mode, enter the `show services ssl proxy statistics` command.

```
user@host> show services ssl proxy statistics

PIC:fwdd0 fpc[0] pic[0]
sessions matched 0
sessions bypassed:non-ssl 0
sessions bypassed:mem overflow 0
sessions bypassed:low memory 0
sessions created 0
sessions ignored 0
sessions active 0
sessions dropped 0
sessions whitelisted 0
whitelisted url category match 0
default profile hit 0
session dropped no default profile 0
policy hit no profile configured 0
```

Meaning

The command output displays the following information:

- Details about the sessions matched for the SSL proxy.

- Details about the default SSL proxy profile such as the sessions where the default profile is applied and the sessions that are dropped due to the absence of the default profile.

SEE ALSO

> *ngfw*

SNI-Based Dynamic Application Information for SSL Proxy Profile

Starting in Junos OS Release 20.4R1, we've enhanced SSL proxy profile selection mechanism by utilizing Server name Indication(SNI) TLS extensions to identify dynamic applications.

SSL proxy module defers SSL profile selection until the dynamic application is detected in a client hello message based on the SNI. After detecting dynamic application, SSL proxy module does a firewall rule lookup based on the identified application and selects an appropriate SSL proxy profile.

Utilizing the SNI-based dynamic application information for SSL proxy profile results in more accurate SSL proxy profile selection for the session. By default, the SNI-based dynamic application information for SSL proxy profile is enabled on the SRX Series Firewall. See *show services ssl proxy counters* to check counters for SSL proxy.

RELATED DOCUMENTATION

> Application Identification Support for Unified Policies | **95**
>
> SSL Proxy | **350**

ICAP Service Redirect

IN THIS SECTION

You can prevent data loss from your network by employing Internet Content Adaptation Protocol (ICAP) redirect services. SRX Series Firewalls support ICAP redirect functionality to redirect HTTP or HTTPS traffic to any third-party server. For more information, read this topic.

Data Loss Prevention (DLP) Using ICAP Service Redirect

IN THIS SECTION

You can prevent data loss from your network by employing Internet Content Adaptation Protocol (ICAP) redirect services. ICAP is a lightweight HTTP-based remote procedure call protocol. ICAP allows its clients to pass HTTP-based content (HTML) to the ICAP servers for performing services such as virus scanning, content translation, or content filtering and so on for the associated client requests.

Junos OS ICAP Support for SRX Series Device

SRX Series Firewalls support ICAP redirect functionality to redirect HTTP or HTTPS traffic to any third-party server. The SRX Series Firewall acts as an SSL proxy server and decrypts the pass-through traffic with the proper SSL profile under a security policy. SRX Series Firewall decrypts HTTPS traffic and redirects HTTP message to a third-party, on-premise server using an ICAP channel. After DLP processing, the traffic is redirected back to the SRX Series Firewall and action is taken according to the results from the ICAP server. If any sensitive data is detected per the policies, the SRX Series Firewall logs, redirects, or blocks the data traffic as configured in the profile.

The following sequences are involved in a typical ICAP redirect scenario:

1. The user opens a connection to a Website on the internet.

2. The request goes through the SRX Series Firewall that is acting as a proxy server.

3. The SRX Series Firewall receives information from the end-host, encapsulates the message and forwards the encapsulated ICAP message to the third-party on-premise ICAP server.

4. The ICAP server receives the ICAP request and analyzes it.

5. If the request does not contain any confidential information, the ICAP server sends it back to the proxy server, and directs the proxy server to send the HTTP to the internet.

6. If the request contains confidential information, you can choose to take action (block, permit, log) as per your requirement.

> **NOTE**: The HTTP throughput depends on the connections between the SRX Series Firewall and the ICAP channel.

Starting in Junos OS Release 19.3R1, ICAP redirect adds `X-Client-IP`, `X-Server-IP`, `X-Authenticated-User`, and `X-Authenticated-Groups` header extensions in an ICAP message to provide information about the source of the encapsulated HTTP message.

ICAP Profile

When you configure ICAP redirect service on SRX Series Firewalls, you must configure the ICAP server information. This profile is applied to a security policy as application services for the permitted traffic. The ICAP profile defines the settings that allow the ICAP server to process request messages, response messages, fallback options (in case of a timeout), connectivity issues, too many requests, or any other conditions.

Service Redirect for Layer 7 Dynamic Applications with Unified Policies

Starting from Junos OS Release 18.2R1, SRX Series Firewalls support ICAP service redirect feature when the device is configured with unified policies.

Unified policies are the security policies that enable you to use dynamic applications as match conditions as part of the existing 5-tuple or 6-tuple (5-tuple with user firewall) match conditions to detect application changes over time.

In a unified policy with dynamic applications as a match condition, you configure an ICAP redirect profile and SSL proxy profile and apply these profiles as application services in the security policy for the permitted traffic. When the traffic matches the policy, the ICAP redirect service profile that is configured as application services is applied. The ICAP server profile defines the behavior of redirection and server specifications. The ICAP server performs the policy scan and the traffic is redirected to the SRX Series Firewall, and the specified action is taken as per the ICAP redirect profile.

Note the following behavior while using ICAP redirect service with unified policy:

- When ICAP redirect is configured in a unified policy and the data that needs to be redirected has arrived and the final policy is not determined, the request is ignored by the ICAP redirect service.

- Because ICAP redirect is one of services located in the service chain, the data received by the ICAP redirect service might be different from the original data. The data sent by the ICAP redirect might affect downstream services.

Benefits of ICAP Redirect Service Support

- Keeps the sensitive data from leaving the network.

- Supports common on-premise server pool for redirection thereby improving management, security, and control of the content.

> **NOTE**: The HTTP throughput depends on the connections between the SRX Series Firewall and SRX ICAP .

Example: Configuring ICAP Redirect Service on SRX Series Firewalls

IN THIS SECTION

This example shows how to define an ICAP redirect profile for an SRX Series Firewall.

Requirements

This example uses the following hardware and software components:

- SRX Series Firewall with Junos OS Release 18.1R1 or later. This configuration example is tested for Junos OS Release 18.1R1.

  ICAP redirect profile for an SRX Series Firewall with unified policies example is tested for Junos OS Release 18.2R1.

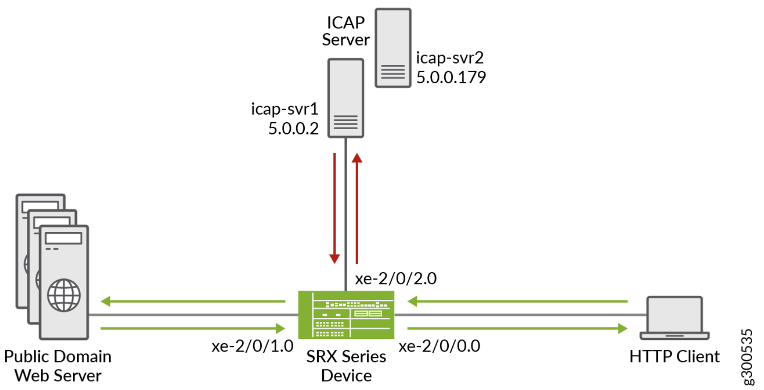
No special configuration beyond device initialization is required before configuring this feature.

Overview

In this example, you configure an ICAP redirect profile and an SSL proxy profile and apply these profiles as application services in the security policy for the permitted traffic.

Figure 18 on page 426 shows the topology used in this example.

Figure 18: ICAP Redirect Topology



To enable the service redirect using ICAP, you must configure an SSL profile to secure the connection to the ICAP server. Next, you configure a security policy to process the traffic, and specify the action for the permitted traffic.

Table 38 on page 426 lists the details of the parameters used in this example.

Table 38: ICAP Redirect Configuration Parameters

Parameters	Names	Description
Profile	icap-pf1	The ICAP server profile allows the ICAP server to process request messages, response messages, fallback options and so on, for the permitted traffic. This profile is applied as an application service in the security policy.
Server name	icap-svr1 icap-svr2	The machine name of the remote ICAP host. Client’s request is redirected to this ICAP server.
Server IP address	5.0.0.2 5.0.0.179	The IP address of the remote ICAP host. Client’s request is redirected to this ICAP server.

Table 38: ICAP Redirect Configuration Parameters *(Continued)*

Parameters	Names	Description
SSL proxy profile	ssl-inspect-profile	An SSL proxy profile defines SSL behavior for the SRX Series Firewall. The SSL proxy profile is applied to the security policy as an application service.
SSL profile	dlp_ssl	The SRX Series Firewall that is acting as an SSL proxy client, initiates and maintains SSL sessions with an SSL server. This configuration enables you to secure the connection to the ICAP server.
Security policy	sp1	In a security policy, apply the SSL proxy profile and ICAP redirect profile. to the permitted traffic.

Configuration

IN THIS SECTION

Procedure

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 services ssl initiation profile dlp_ssl trusted-ca all
set services ssl initiation profile dlp_ssl actions ignore-server-auth-failure
set services ssl initiation profile dlp_ssl actions crl disable
set services icap-redirect profile icap-pf1 server icap-svr1 host 5.0.0.2
set services icap-redirect profile icap-pf1 server icap-svr1 reqmod-uri echo
set services icap-redirect profile icap-pf1 server icap-svr1 respmod-uri echo
set services icap-redirect profile icap-pf1 server icap-svr1 sockets 64
set services icap-redirect profile icap-pf1 server icap-svr2 host 5.0.0.179
set services icap-redirect profile icap-pf1 server icap-svr2 reqmod-uri echo
set services icap-redirect profile icap-pf1 server icap-svr2 respmod-uri echo
set services icap-redirect profile icap-pf1 server icap-svr2 sockets 64
set services icap-redirect profile icap-pf1 server icap-svr2 tls-profile dlp_ssl
set services icap-redirect profile icap-pf1 http redirect-request
set services icap-redirect profile icap-pf1 http redirect-response
set security policies from-zone trust to-zone untrust policy sec_policy match source-address any
set security policies from-zone trust to-zone untrust policy sec_policy match destination-
address any
set security policies from-zone trust to-zone untrust policy sec_policy match application any
set security policies from-zone trust to-zone untrust policy sec_policy then permit application-
services ssl-proxy profile-name ssl-inspect-profile
set security policies from-zone trust to-zone untrust policy sec_policy then permit application-
services icap-redirect icap-pf1
set security policies default-policy permit-all
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 xe-2/0/0.0
set security zones security-zone trust interfaces xe-2/0/2.0
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 xe-2/0/1.0
set interfaces xe-2/0/0 unit 0 family inet address 192.0.2.1/24
set interfaces xe-2/0/0 unit 0 family inet6 address 2001:db8::1/64
set interfaces xe-2/0/1 unit 0 family inet address 198.51.100.1/24
set interfaces xe-2/0/1 unit 0 family inet6 address 2001:db8::2/64
set interfaces xe-2/0/2 unit 0 family inet address 198.51.100.2/24
set interfaces xe-2/0/2 unit 0 family inet6 address 2001:db8::3/64
```

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 ICAP redirect service:

1. Configure the SSL profile for a secured connection with the ICAP server.

   ```
   [edit services]
   user@host# set ssl initiation profile dlp_ssl trusted-ca all
   ```

```
user@host# set ssl initiation profile dlp_ssl actions ignore-server-auth-failure
user@host# set ssl initiation profile dlp_ssl actions crl disable
```

2. Configure the ICAP redirect profile for the first server (icap-svr1).

```
[edit services]
user@host# set icap-redirect profile icap-pf1 server icap-svr1 host 5.0.0.2
user@host# set icap-redirect profile icap-pf1 server icap-svr1 reqmod-uri echo
user@host# set icap-redirect profile icap-pf1 server icap-svr1 respmod-uri echo
user@host# set icap-redirect profile icap-pf1 server icap-svr1 sockets 64
```

3. Configure the ICAP redirect profile for the second server (icap-svr2).

```
[edit services]
user@host# set icap-redirect profile icap-pf1 server icap-svr2 host 5.0.0.179
user@host# set icap-redirect profile icap-pf1 server icap-svr2 reqmod-uri echo
user@host# set icap-redirect profile icap-pf1 server icap-svr2 respmod-uri echo
user@host# set icap-redirect profile icap-pf1 server icap-svr2 sockets 64
user@host# set icap-redirect profile icap-pf1 server icap-svr2 tls-profile dlp_ssl
```

4. Configure the redirect request and the redirect response for the HTTP traffic.

```
[edit services]
user@host# set icap-redirect profile icap-pf1 http redirect-request
user@host# set icap-redirect profile icap-pf1 http redirect-response
```

5. Configure a security policy to apply application services for the ICAP redirect to the permitted traffic.

```
[edit security]
user@host# set policies from-zone trust to-zone untrust policy sec_policy match source-
address any
user@host# set policies from-zone trust to-zone untrust policy sec_policy match destination-
address any
user@host# set policies from-zone trust to-zone untrust policy sec_policy match application
any
user@host# set policies from-zone trust to-zone untrust policy sec_policy then permit
application-services ssl-proxy profile-name ssl-inspect-profile
user@host# set policies from-zone trust to-zone untrust policy sec_policy then permit
```

```
application-services icap-redirect icap-pf1
user@host# set policies default-policy permit-all
```

6. Configure interfaces and zones.

```
[edit]
user@host# set interfaces xe-2/0/0 unit 0 family inet address 192.0.2.1/24
user@host# set interfaces xe-2/0/0 unit 0 family inet6 address 2001:db8::1/64
user@host# set interfaces xe-2/0/1 unit 0 family inet address 198.51.100.1/24
user@host# set interfaces xe-2/0/1 unit 0 family inet6 address 2001:db8::2/64
user@host# set interfaces xe-2/0/2 unit 0 family inet address 198.51.100.2/24
user@host# set interfaces xe-2/0/2 unit 0 family inet6 address 2001:db8::3/64
user@host# set zones security-zone trust host-inbound-traffic system-services all
user@host# set zones security-zone trust host-inbound-traffic protocols all
user@host# set zones security-zone trust interfaces xe-2/0/0.0
user@host# set zones security-zone trust interfaces xe-2/0/2.0
user@host# set zones security-zone untrust host-inbound-traffic system-services all
user@host# set zones security-zone untrust host-inbound-traffic protocols all
user@host# set zones security-zone untrust interfaces xe-2/0/1.0
```

Results

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

```
user@host# show services ssl
    initiation {
        profile dlp_ssl {
            trusted-ca all;
            actions {
                ignore-server-auth-failure;
                crl {
                    disable;
                }
            }
        }
    }
```

```
user@host#  show services icap-redirect
profile icap-pf1 {
    server icap-svr1 {
        host 5.0.0.2;
        reqmod-uri echo;
        respmod-uri echo;
        sockets 64;
    }
        server icap-svr2 {
            host 5.0.0.179;
            reqmod-uri echo;
            respmod-uri echo;
            sockets 10;
            tls-profile dlp_ssl;
        }
    http {
        redirect-request;
        redirect-response;
    }
}
```

```
user@host#  show security policies
    from-zone trust to-zone untrust {
        policy sec_policy {
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit {
                    application-services {
                        ssl-proxy {
                            profile-name ssl-inspect-profile;
                        }
                        icap-redirect icap-pf1;
                    }
                }
            }
        }
    }
    default-policy {
        permit-all;
    }
```

```
user@host#  show security zones
security-zone trust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        xe-2/0/0.0;
        xe-2/0/2.0;
    }
}
security-zone untrust {
    host-inbound-traffic {
        system-services {
            all;
        }
        protocols {
            all;
        }
    }
    interfaces {
        xe-2/0/1.0;
    }
}
```

```
user@host#  show interfaces
xe-2/0/0 {
    unit 0 {
        family inet {
            address 192.0.2.1/24;
        }
        family inet6 {
            address 2001:db8::1/64;
        }
    }
}
xe-2/0/1 {
    unit 0 {
        family inet {
            address 198.51.100.1/24;
        }
        family inet6 {
            address 2001:db8::2/64;
        }
    }
}
xe-2/0/2 {
    unit 0 {
        family inet {
            address 198.51.100.2/24;
        }
        family inet6 {
            address 2001:db8::3/64;
        }
    }
}
```

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

Configuring ICAP Service Redirect for Unified Policy

Step-by-Step Procedure

You can follow the procedure below if you have configured a unified policy (supported from Junos OS Release 18.2R1).

The following example requires you to navigate to 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 ICAP redirect service:

1. Configure the SSL profile for secured connection with the ICAP server.

```
[edit services]
user@host# set ssl initiation profile dlp_ssl trusted-ca all
user@host# set ssl initiation profile dlp_ssl actions ignore-server-auth-failure
user@host# set ssl initiation profile dlp_ssl actions crl disable
```

2. Configure the ICAP redirect profile for the first server (icap-svr1).

```
[edit services]
user@host# set icap-redirect profile icap-pf1 server icap-svr1 host 5.0.0.2
user@host# set icap-redirect profile icap-pf1 server icap-svr1 reqmod-uri echo
user@host# set icap-redirect profile icap-pf1 server icap-svr1 respmod-uri echo
user@host# set icap-redirect profile icap-pf1 server icap-svr1 sockets 64
```

3. Configure the ICAP redirect profile for the second server (icap-svr2).

```
[edit services]
user@host# set icap-redirect profile icap-pf1 server icap-svr2 host 5.0.0.179
user@host# set icap-redirect profile icap-pf1 server icap-svr2 reqmod-uri echo
user@host# set icap-redirect profile icap-pf1 server icap-svr2 respmod-uri echo
user@host# set icap-redirect profile icap-pf1 server icap-svr2 sockets 64
user@host# set icap-redirect profile icap-pf1 server icap-svr2 tls-profile dlp_ssl
```

4. Configure the redirect request for HTTP traffic.

```
[edit services]
user@host# set icap-redirect profile icap-pf1 http redirect-request
user@host# set icap-redirect profile icap-pf1 http redirect-response
```

5. Configure a security policy to apply application services for the ICAP redirect to the permitted traffic.

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

```
any
user@host# set policies from-zone trust to-zone untrust policy sec_policy match dynamic-
application junos:HTTP
user@host# set policies from-zone trust to-zone untrust policy sec_policy then permit
application-services ssl-proxy profile-name ssl-inspect-profile
user@host# set policies from-zone trust to-zone untrust policy sec_policy then permit
application-services icap-redirect icap-pf1
user@host# set policies default-policy permit-all
```

Verification

IN THIS SECTION

Verifying ICAP Redirect Configuration

Purpose

Verify that the ICAP redirect service is configured on the device.

Action

From operational mode, enter the `show services icap-redirect status` and `show services icap-redirect statistic` commands.

```
user@host> show services icap-redirect status

ICAP Status :
         Spu-1 Profile: icap-pf1 Server: icap-svr1 : UP
ICAP Status :
         Spu-1 Profile: icap-pf1 Server: icap-svr2 : UP
ICAP Status :
         Spu-2 Profile: icap-pf1 Server: icap-svr1 : UP
ICAP Status :
         Spu-2 Profile: icap-pf1 Server: icap-svr2 : UP
ICAP Status :
         Spu-3 Profile: icap-pf1 Server: icap-svr1 : UP
```

```
ICAP Status :
         Spu-3 Profile: icap-pf1 Server: icap-svr2 : UP
user@host> show services icap-redirect statistic

ICAP Redirect statistic:
  Message Redirected                  : 2
      Message REQMOD Redirected       : 1
      Message RESPMOD Redirected      : 1
  Message Received                    : 2
      Message REQMOD Received         : 1
      Message RESPMOD Received        : 1
Fallback:      permit           log-permit       reject
  Timeout         0                 0                 0
  Connectivity    0                 0                 0
  Default         0                 0                 0
```

Meaning

The status Up indicates that the ICAP redirect service is enabled. The Message Redirected and the Message Received fields show the number of HTTP requests that have passed through the ICAP channel.

Release History Table

Release	Description
19.3R1	Starting in Junos OS Release 19.3R1, ICAP redirect adds X-Client-IP, X-Server-IP, X-Authenticated-User, and X-Authenticated-Groups header extensions in an ICAP message to provide information about the source of the encapsulated HTTP message.
18.2R1	Starting from Junos OS Release 18.2R1, SRX Series Firewalls support ICAP service redirect feature when the device is configured with unified policies

SSL Decryption Mirroring

IN THIS SECTION

SSL decryption mirroring feature enables you to monitor SSL decrypted application traffic entering and exiting the SRX Series Firewall. For more information on SSL decryption mirroring, read this topic.

Understanding SSL Decryption Mirroring Functionality

IN THIS SECTION

Starting in Junos OS Release 18.4R1, SSL decryption mirroring functionality for SSL forward proxy and for SSL reverse proxy is introduced.

SSL decryption mirroring feature enables you to monitor SSL decrypted application traffic entering and exiting the SRX Series Firewall. When you enable this feature, the SRX Series Firewall uses an Ethernet interface—the configured SSL decryption mirroring interface—to forward a copy of the decrypted SSL traffic to a trusted traffic collection tool or a network analyzer for inspection and analysis. Typically, you connect this external monitoring device to the SSL decryption mirroring interface through a switching device. The external mirror traffic collector port is the port (or interface) that receives the copy of the decrypted traffic from the SSL decryption mirroring interface on the SRX Series Firewall.

To use the SSL decryption mirroring feature, you define an SSL proxy profile, and apply it to the security policy. The security policy rule allows you to define traffic that you want the device to decrypt. When you attach the SSL proxy profile to the security policy rule, the traffic matching the security policy rule is decrypted. The SSL decryption mirroring interface delivers a copy of decrypted HTTPS and STARTTLS (POP3S/SMTPS/IMAPS) traffic to a trusted external device or traffic collection tool for inspection and analysis.

The embedded 5-tuple data of the decrypted IP packet includes the same following values as the encrypted IP packets:

- Source IP address
- Destination IP address
- Source port number
- Destination port number
- Protocol number

Retaining the same 5-tuple data without reconfiguration ensures that the decrypted traffic is saved in packet-capturing format (Wireshark) and you can replay the data later.

Only TCP sequence numbers and ACK numbers are constructed based on the actual decrypted payload forwarded on the SSL decryption mirroring port. If the decrypted packet size exceeds the maximum transmission unit (MTU) size of the SSL decryption mirroring port, then the decrypted payload is divided into multiple TCP segments based on the MTU size requirements.

SSL Decryption Mirroring Before or After Policy Enforcement

By default, the SRX Series Firewall forwards the SSL decrypted payload to the mirror port before Junos OS enforces Layer 7 security services, including IDP, Juniper ATP Cloud, and Content Security. This option allows you to replay events and analyze traffic that generates a threat or triggers a drop action.

You can also configure mirroring of the decrypted traffic after enforcing the security policy. With this option, only traffic that is forwarded through the security policy is mirrored. However, if the decrypted payload is modified while enforcing the security policy, the modified decrypted payload is forwarded on the mirror port. Similarly, if the decrypted traffic is dropped because of policy enforcement (for example, when a threat is detected in the decrypted traffic), that particular decrypted traffic is not forwarded on the mirror port.

SSL Decryption Mirroring Support

- Supported for SSL forward proxy and SSL reverse proxy.
- Supported for both IPv4 and IPv6 traffic.

- The SSL decrypted traffic available on the mirror port is in cleartext format. All the cipher suites that are supported by SSL proxy support SSL decryption mirroring functionality. For the list of supported cipher suites, see SSL Proxy Overview.

Benefits of SSL Decryption Mirroring

- Enables comprehensive data capture for auditing, forensic investigations, and historical purposes.
- Provides data leak prevention.
- Enables additional security processing done by third-party appliances for IDP, Content Security, and so on.
- Provides insight about the threats involved.

Limitations

- SSL decryption mirroring cannot be configured on the st0 tunnel interface.

SSL Decryption Mirroring Support in Chassis Cluster

Starting in Junos OS Release 18.4R1-S2 and Junos OS Release 19.2R1, the SSL decryption mirroring feature is supported on redundant Ethernet (reth) interface on SRX Series Firewalls operating in a chassis cluster.

```
set interfaces reth20 redundant-ether-options redundancy-group 1
 set interfaces reth20 unit 0 family inet
```

Configuring SSL Decryption Mirroring

IN THIS SECTION

This example shows how to enable mirroring of SSL decrypted traffic on an SRX Series Firewall.

Configuration

Step-by-Step Procedure

Use the following steps to configure the SSL decryption mirroring.

1. Define the SSL decryption mirroring interface with logical unit number 0.

```
user@host# set interfaces ge-0/0/2 unit 0
```

2. Specify the SSL decryption mirroring interface in the SSL proxy profile.

```
user@host# set services ssl proxy profile profile-1 mirror-decrypt-traffic interface ge-0/0/2.0
```

   Ge-0/0/2.0 is configured as designated SSL decryption mirroring interface.

3. Specify the MAC address of the of the external mirror traffic collector port.

```
user@host# set services ssl proxy profile profile-1 mirror-decrypt-traffic destination-mac-
address 00:50:56:a6:5f:1f
```

4. Create a security policy by specifying the match criteria for the traffic.

```
user@host# set security policies from-zone trust to-zone untrust policy policy-1 match source-
address any
```

```
user@host# set security policies from-zone trust to-zone untrust policy policy-1 match
destination-address any
```

```
user@host# set security policies from-zone trust to-zone untrust policy policy-1 match
application any
```

5. Attach the SSL proxy profile to the security policy rule.

   ```
   user@host# set security policies from-zone trust to-zone untrust policy policy-1 then permit
   application-services ssl-proxy profile-name profile-1
   ```

   This configuration enables the external mirror traffic collector port (or interface) to receive the copy of the decrypted traffic from the SSL decryption mirroring interface on the SRX Series Firewall.

Results

From configuration mode, confirm your configuration by entering the `show services ssl proxy profile` and `show security policies from-zone trust to-zone untrust policy` commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show services ssl proxy profile profile-1
server-certificate Email_server_cert;
    mirror-decrypt-traffic {
    interface ge-0/0/2.0;
    destination-mac-address 00:50:56:a6:5f:1f;
}
```

```
[edit]
user@host# show security policies from-zone trust to-zone untrust policy policy-1
match {
    source-address any;
    destination-address any;
    application any;
}
then {
    permit {
        application-services {
            ssl-proxy {
                profile-name profile-1;
            }
        }
    }
}
```

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

Requirements

This example uses the following hardware and software components:

- Any SRX Series Firewall with Junos OS Release 18.4R1 or later. This configuration example is tested for Junos OS Release 18.4R1.

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

Before you begin:

- Configure SSL proxy. See SSL Proxy Overview.
- The SSL decryption mirroring interface that you configure doesn't need to be part of any security zones.
- Ensure that SSL decryption mirroring interface and the actual client-server SSL traffic processing interfaces are part of the same routing instance.
- Ensure that the SSL decryption mirroring interface on the SRX Series Firewall and the external mirror traffic collector port must be part of the same broadcast domain.

NOTE: You don't need to configure a separate security policy to allow traffic from SRX Series Firewall to the SSL decryption mirroring interface..

Overview

In this example, configure an SSL forward proxy profile by specifying the name of the SSL decryption mirroring interface and the MAC address of the external mirror traffic collector port. Next, create a security policy and invoke the SSL proxy as application service on the permitted traffic. The traffic matching the security policy rule is decrypted. A copy of the decrypted SSL payload is then encapsulated into an IP packet and forwarded to the on the external mirror traffic collector port through SSL decryption mirroring interface.

Figure 19 on page 443 illustrates the topology used in this example.

Figure 19: SSL Decryption Mirroring

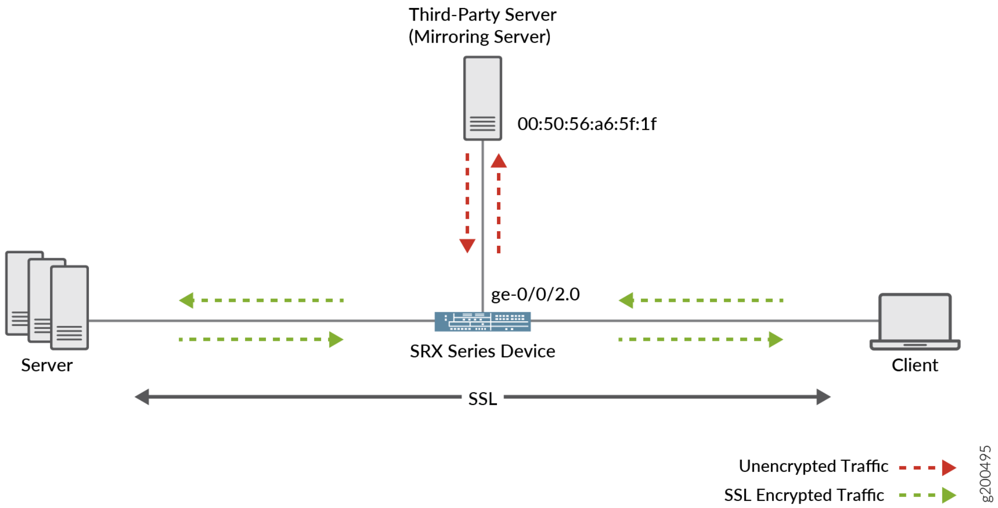


Table 39 on page 443 provides the details of the parameters used in this example.

Table 39: Parameters Used in SSL Decryption Mirroring Example

Parameter	Name
SSL decryption mirroring interface on SRX Series Firewall	ge-0/0/2.0
MAC address of the external mirror traffic collector port	00:50:56:a6:5f:1f
SSL proxy profile	profile-1
Security policy	policy 1

Verification

IN THIS SECTION

Verify SSL Proxy Configuration

Purpose

Confirm that the configuration is working properly by displaying the SSL proxy statistics.

Action

From operational mode, enter the `show services ssl proxy statistics` command.

```
user@host> show services ssl proxy statistics
PIC:fwdd0 fpc[0] pic[0]
sessions matched 30647
sessions bypassed:non-ssl 0
sessions bypassed:mem overflow 0
sessions bypassed:low memory 0
sessions created 25665
sessions ignored 0
sessions active 0
sessions dropped 0
sessions whitelisted 0
whitelisted url category match 0
default profile hit 0
session dropped no default profile 0
policy hit no profile configured 0
```

SEE ALSO

mirror-decrypt-traffic

Release History Table

Release	Description
18.4R1	Starting in Junos OS Release 18.4R1, SSL decryption mirroring functionality for SSL forward proxy and for SSL reverse proxy is introduced

SSL Proxy Logs

IN THIS SECTION

SSL Proxy Logs

IN THIS SECTION

SSL Proxy Logs

When logging is enabled in an SSalphaTable 40 on page 445.

Table 40: SSL Proxy Logs

Syslog Type	Description
SSL_PROXY_SSL_SESSION_DROP	Logs generated when a session is dropped by SSL proxy.
SSL_PROXY_SSL_SESSION_ALLOW	Logs generated when a session is processed by SSL proxy even after encountering some minor errors.
SSL_PROXY_SESSION_IGNORE	Logs generated if non-SSL sessions are initially mistaken as SSL sessions.
SSL_PROXY_SESSION_WHITELIST	Logs generated when a session is allowlisted.

Table 40: SSL Proxy Logs *(Continued)*

Syslog Type	Description
SSL_PROXY_ERROR	Logs used for reporting errors.
SSL_PROXY_WARNING	Logs used for reporting warnings.
SSL_PROXY_INFO	Logs used for reporting general information.

You we can use SSL_PROXY_SESSION_WHITELIST and SSL_PROXY_INFO logs to check the URLs logged in. Example:

```
For non-whitelisted session -
SSL_PROXY_INFO [junos@2636.1.1.1.2.129 logical-system-name="root-logical-system" session-id="17"
source-address="5.0.0.1" source-port="57558" destination-address="4.0.0.1" destination-
port="10302" nat-source-address="5.0.0.1" nat-source-port="57558" nat-destination-
address="4.0.0.1" nat-destination-port="10302" profile-name="ssl-inspect-profile" source-zone-
name="trust" source-interface-name="ge-0/0/0.0" destination-zone-name="untrust" destination-
interface-name="ge-0/0/1.0" message="NA" sni="www.facebook.com" url-category="NULL"]
```

```
For whitelisted session -
SSL_PROXY_SESSION_WHITELIST [junos@2636.1.1.1.2.129 logical-system-name="root-logical-system"
session-id="18" url="4.0.0.1" source-address="5.0.0.1" source-port="57560" destination-
address="4.0.0.1" destination-port="10302" nat-source-address="5.0.0.1" nat-source-port="57560"
nat-destination-address="4.0.0.1" nat-destination-port="10302" profile-name="ssl-inspect-
profile" source-zone-name="trust" source-interface-name="ge-0/0/0.0" destination-zone-
name="untrust" destination-interface-name="ge-0/0/1.0" message="session whitelisted url category
match SNI www.youtube.com URL_CATEGORY CATEGORY-1"]
```

Check System Log Explorer for more details.

All logs contain similar information as shown in the following example (actual order of appearance):

```
logical-system-name, session-id, source-ip-address, source-port, destination-ip-
address,destination-port,
nat-source-ip-address, nat-source-port, nat-destination-ip-address, nat-destination-port, proxy
```

```
profile name, source-zone-name,
source-interface-name, destination-zone-name,destination-interface-name, message
```

The `message` field contains the reason for the log generation. One of three prefixes shown in Table 41 on page 447 identifies the source of the message. Other fields are descriptively labeled.

Table 41: SSL Proxy Log Prefixes

Prefix	Description
system	Logs generated due to errors related to the device or an action taken as part of the SSL proxy profile. Most logs fall into this category.
openssl error	Logs generated during the handshaking process if an error is detected by the openssl library.
certificate error	Logs generated during the handshaking process if an error is detected in the certificate (x509 related errors).

Sample logs:

```
Jun  1 05:11:13 4.0.0.254 junos-ssl-proxy: SSL_PROXY_SSL_SESSION_DROP: lsys:root  23 <
203.0.113.1/35090->192.0.2.1/443>  NAT:< 203.0.113.1/35090->192.0.2.1/443> ssl-inspect-profile
<untrust:ge-0/0/0.0->trust:ge-0/0/1.0> message:certificate error: self signed certificate
```

NOTE: These logs capture sessions that are dropped by SSL proxy, not sessions that are marked by other modules that also use SSL proxy services.

For SSL_PROXY_SESSION_WHITELIST messages, an additional `host` field is included after the `session-id` and contains the IP address of the server or domain that has been allowlisted.

```
Jun  1 05:25:36 4.0.0.254 junos-ssl-proxy: SSL_PROXY_SESSION_WHITELIST: lsys:root 24
host:192.0.2.1/443<203.0.113.1/35090->192.0.2.1/443>  NAT:< 203.0.113.1/35090->192.0.2.1/443 >
ssl-inspect-profile  <untrust:ge-0/0/0.0->trust:ge-0/0/1.0> message:system: session whitelisted
```

Enabling Debugging and Tracing for SSL Proxy

Debug tracing on both Routing Engine and the Packet Forwarding Engine can be enabled for SSL proxy by setting the following configuration:

```
user@host# set services ssl traceoptions file file-name
```

SSL proxy is supported on SRX340, SRX345, SRX380, SRX550M, SRX1500, SRX4100, SRX4200, SRX5400, SRX5600, SRX5800 devices and vSRX Virtual Firewall instances. Table 42 on page 448 shows the supported levels for trace options.

Table 42: Trace Levels

Cause Type	Description
Brief	Only error traces on both the Routing Engine and the Packet Forwarding Engine.
Detail	Packet Forwarding Engine–Only event details up to the handshake should be traced. Routing Engine–Traces related to commit. No periodic traces on the Routing Engine will be available
Extensive	Packet Forwarding Engine–Data transfer summary available. Routing Engine–Traces related to commit (more extensive). No periodic traces on the Routing Engine will be available.
Verbose	All traces are available.

Table 43 on page 448 shows the flags that are supported.

Table 43: Supported Flags in Trace

Cause Type	Description
cli-configuration	Configuration-related traces only.

Table 43: Supported Flags in Trace *(Continued)*

Cause Type	Description
initiation	Enable tracing on the SSL-I plug-in.
proxy	Enable tracing on the SSL-Proxy-Policy plug-in.
termination	Enable tracing on the SSL-T plug-in.
selected-profile	Enable tracing only for profiles that have **enable-flow-tracing** set.

You can enable logs in the SSL proxy profile to get to the root cause for the drop. The following errors are some of the most common:

- Server certification validation error. Check the trusted CA configuration to verify your configuration.
- System failures such as memory allocation failures.
- Ciphers do not match.
- SSL versions do not match.
- SSL options are not supported.
- Root CA has expired. You need to load a new root CA.

You can enable the **ignore-server-auth-failure** option in the SSL proxy profile to ensure that certificate validation, root CA expiration dates, and other such issues are ignored. If sessions are inspected after the **ignore-server-auth-failure** option is enabled, the problem is localized.

SEE ALSO

traceoptions (Services SSL)

Operational Commands to Troubleshoot SSL Sessions

IN THIS SECTION

In the CLI, the operational commands provide information that can help with troubleshooting. You can use show commands to determine and analyze the statistical counters and metrics related to any traffic loss and take an appropriate corrective measure. This topic covers information for monitoring, displaying, and verifying of SSL-related issues using the operational mode commands.

Displaying Active SSL Sessions

IN THIS SECTION

Purpose

Display information about all the active SSL sessions on the device.

Action

Use the **show security flow session ssl** command.

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

Output:
```

```
Session ID: 1, Policy name: default-permit/5, Timeout: 1746, Valid
In: 4.0.0.1/37369 --> 5.0.0.1/4433;tcp, Conn Tag: 0x0, If: xe-0/0/0.0, Pkts: 6, Bytes: 671,
Out: 5.0.0.1/4433 --> 4.0.0.1/37369;tcp, Conn Tag: 0x0, If: xe-0/0/1.0, Pkts: 7, Bytes: 1635,
```

Meaning

The output shows all standard flow information including the session ID, timeout value for the session, the direction of the flow, the source address and port, the destination address and port, the IP protocol, and the interface used for the session. Example:

- The policy name that allowed this traffic is default-permit.
- The timeout value.
- Both the source IP and the destination IP are displayed with their respective source/destination ports.
- Session type.
- The source interface and the destination interface for this session.

For details about the output fields of the command, see *show security flow session ssl*.

Displaying Active SSL Sessions Details

IN THIS SECTION

Purpose

Display detail information about the active SSL sessions on the device.

Action

From the operational mode, use the **show security flow session extensive ssl** command.

```
user@host > show security flow session extensive ssl
Output:
Session ID: 1, Status: Normal
Flags: 0x42/0x20000000/0x2/0x10103
Policy name: 1/5
Source NAT pool: Null
Dynamic application: junos:UNKNOWN,
Encryption: Unknown
Application traffic control rule-set: INVALID, Rule: INVALID
Maximum timeout: 1800, Current timeout: 1636
Session State: Valid
Start time: 587131, Duration: 163
In: 4.0.0.1/37369 --> 5.0.0.1/4433;tcp,
Conn Tag: 0x0, Interface: xe-0/0/0.0,
Session token: 0x7, Flag: 0x2621
Route: 0xa0010, Gateway: 4.0.0.1, Tunnel: 0
Port sequence: 0, FIN sequence: 0,
FIN state: 0,
Pkts: 6, Bytes: 671
Out: 5.0.0.1/4433 --> 4.0.0.1/37369;tcp,
Conn Tag: 0x0, Interface: xe-0/0/1.0,
Session token: 0x8, Flag: 0x2620
Route: 0xb0010, Gateway: 5.0.0.1, Tunnel: 0
Port sequence: 0, FIN sequence: 0,
FIN state: 0,
Pkts: 7, Bytes: 1635
Total sessions: 1
```

Meaning

The output of the command displays extensive information about all the active sessions on the device.

Display information includes the session ID, the Network Address Translation (NAT) source pool (if source NAT is used), the configured timeout value for the session and its standard timeout, and the session start time and how long the session has been active, direction of the flow, the source address and port, the destination address and port, the IP protocol, and the interface used for the session.

Example:

- The policy name that allowed this traffic is default-permit.
- The maximum timeout and current timeout values.
- Session type.
- The source interface and the destination interface for the session
- The next-hop gateway IP address
- AppQoS rule set details.

For details about the output fields of the command, see *show services ssl session*.

Displaying Specific SSL Session Details

IN THIS SECTION

Purpose

Display information about the specific SSL session.

Action

Use the **show services ssl session 56** command.

```
Lsys Name : root-logical-system

PIC:fpc0 fpc[0] pic[0] ------

Session ID           : 56
Connection Type      : PROXY
SSL Profile          : SSL_PROFILE
Resumed Session      : No
```

```
One-crypto          : Disabled
Async-crypto        : Enabled
Renegotiation count : 0
Server Certificate Subject Name    : /C=IN/ST=KA/L=BNG/O=JN/OU=XYZ/CN=server/emailAddress=ser
Server Cert verification status    : OK
CRL check           : Enabled
Action              : Allow
SSL_T Details :

      Key size      : 2048
      cipher        : ECDHE-RSA-AES256-GCM-SHA384
      TLS version   : 1.2
SSL_I Details :

      Key size      : 2048
      Cipher        : ECDHE-RSA-AES256-GCM-SHA384
      TLS version   : 1.2
```

Meaning

You can get the detail information about the specific SSL session with this command. Example:

- Session ID, connection type and SSL profile used for the session.
- Server certificate subject name and verification status.
- CRL check status and action.
- SSL Initiation and termination details.
- The source interface and the destination interface for this session.

For details about the output fields of the command, see *show security flow session ssl*.

Display SSL Certificates

IN THIS SECTION

Purpose

Display the digital certificates available on the device.

Action

From the operational mode, use the **show services ssl certificate all** command.

```
user@host > show services ssl certificate all

Lsys Name : root-logical-system
PIC:fwdd0 fpc[0] pic[0] ------
CertId
-----------------------------
ssl-inspect-ca
ssl-cert-4k
```

Meaning

Display the list of all SSL certificates active on the device. SSL sessions use these certificates to establish a secure communication between a client and a server.

For details about the output fields of the command, see *show services ssl certificate*.

Display SSL Certificate Information

IN THIS SECTION

Purpose

Display brief information about the SSL certificate.

Action

From the operational mode, use the **show services ssl certificate brief certificate-id *certificate-identifier*** command. Following samples show command outputs for CA certificate and local certificates.

```
user@host > show services ssl certificate brief certificate-id trusted-ca

Lsys Name : root-logical-system

PIC:fpc0 fpc[0] pic[0] ------

CertID : trusted-ca
Certificate Type : CA-CERT
Issuer : /C=IN/ST=KA/L=BNG/O=XYZ/OU=ABC/CN=5.0.0.1/emailAddress=newca@test.com
Subject : /C=IN/ST=KA/L=BNG/O=XYZ/OU=ABC/CN=5.0.0.1/emailAddress=newca@test.com
Public Key algorithm : rsaEncryption
```

```
user@host> show services ssl certificate brief certificate-id ssl-inspect-ca

Lsys Name : root-logical-system

PIC:fpc0 fpc[0] pic[0] ------

CertID                : ssl-inspect-ca
Certificate Type      : LOCAL-CERT
Issuer                : /DC=dc/CN=xyz.com/OU=IT/O=abc/L=bng/ST=KA/C=IN
Subject               : /DC=dc/CN=xyz.com/OU=IT/O=abc/L=bng/ST=KAC=IN
Validity :
    Not before        : Mon 02/18/2019 07:30:37 AM
    Not after         : Sat 02/17/2024 07:30:37 AM
Public Key algorithm  : rsaEncryption
```

Meaning

Displays details about the certificate including certificate ID, type, issuer of the certificate, and encryption algorithm used. The `type` field displays the type of the certificate—That is—CA-CERT or LOCAL-CERT. CA-Cert certificate is an authorized certificate issued by trusted certificate authority and LOCAL-CERT is a self-signed certificate.

Note that the output of the commands vary depending on the type of certificate.

For details about the output fields of the command, see *show services ssl certificate*.

Display SSL Certificate Details

IN THIS SECTION

Purpose

Display detail information about the SSL certificate.

Action

From the operational mode, use the **show services ssl certificate detail** ***certificate-identifier*** command.

```
user@host > show services ssl certificate detail certificate-id ssl-inspect-ca
Lsys Name : root-logical-system

PIC:fpc0 fpc[0] pic[0] ------
```

```
CertID                : ssl-inspect-ca
Certificate Type      : LOCAL-CERT
cert modify time      : Mon 02/18/2019 07:30:37 AM
key modify time       : Mon 02/18/2019 07:30:23 AM
certificate version   : 3
serial number         : 72 a4 a8 12 0e a0 da 5f ee 27 47 d8 19 7c 76 b5
Issuer                : /DC=dc/CN=xyz.com/OU=IT/O=xyz/L=blr/ST=KA/C=IN
Subject               : /DC=dc/CN=xyz.com/OU=IT/O=xyz/L=blr/ST=KA/C=IN
Validity :
    Not before        : Mon 02/18/2019 07:30:37 AM
    Not after         : Sat 02/17/2024 07:30:37 AM
Public Key algorithm  : rsaEncryption
Signature Algorithm   : sha256WithRSAEncryption
```

```
user@host > show services ssl certificate detail certificate-id test
Lsys Name : root-logical-system

PIC:fpc0 fpc[0] pic[0] ------

CertID                : test
Certificate Type      : CA-CERT
cert modify time      : Mon 09/02/2019 09:47:48 PM
certificate version   : 1
serial number         : 21 a8 d6 00 eb 24 1f 78 9a e5 0e ec 6a 39 ce 65 66 42 8c 0a
Issuer                : /C=IN/ST=KA/L=BLR/O=XYZ/OU=ABC/CN=5.0.0.1/emailAddress=newca@test.com
Subject               : /C=IN/ST=KA/L=BLR/O=XYZ/OU=ABC/CN=5.0.0.1/emailAddress=newca@test.com
Public Key algorithm  : rsaEncryption
Signature Algorithm   : sha256WithRSAEncryption
CRL :
    present           : no
    check             : enabled
    download-failed   : true
    check-on-download-fail : enabled
```

Meaning

Displays details about the certificate including certificate ID, type, last modified date, version, serial number, issuer, subject, validity, and encryption algorithm used.

Example:

- Type of the certificate. The `type` field displays the type of the certificate—That is—CA-CERT or LOCAL-CERT. CA-Cert certificate is an authorized certificate issued by trusted certificate authority and LOCAL-CERT is a self-signed certificate.
- Subject and issuer of the certificate.
- Certificate validity from-date and to-date.
- Public key algorithms used.
- Algorithm used by the certificate authority to sign the certificate.
- CRL-related updates (CA certificates only)

For details about the output fields of the command, see *show services ssl certificate*.

SSL Proxy Counters All

IN THIS SECTION

Purpose

Display all the statistical counters for the SSL proxy sessions.

Action

From the operational mode, use the **show services ssl proxy counters all** command.

```
user@host > show services ssl proxy counters all
Lsys Name : root-logical-system

PIC:fpc0 fpc[0] pic[0] ------

session create failed                    0
```

```
non SSL sessions recieved                      0
Memory failures                                0
session dropped                                0
sessions matched                               0
sessions created                               0
sessions destroyed                             0
sessions ignored                               0
sessions ignored : backup only                 0
sessions whitelisted : IP based                0
sessions whitelisted : url based               0
crl : data added                               0
crl : certificate revoked                      0
crl : no crl info present                      0
crl : no CA certificate                        0
SSL sessions                                   0
SMTP over STARTTLS                             0
IMAP over STARTTLS                             0
POP3 over STARTTLS                             0
SMTP  sessions                                 0
IMAP  sessions                                 0
POP3  sessions                                 0
Server not supporting STARTTLS                 0
Client not supporting STARTTLS                 0
Unified policy : default profile hit           0
Unified policy : no default profile            0
```

Meaning

The output display the counters details related to SSL proxy sessions. These counters generally increment whenever there is some activity such as session matched, session created, and so on.

Example:

- Count of sessions created, matched, ignored or destroyed.
- Number of sessions allowlisted based on IP address and URL categories.
- Session counts based on CRL-related information such as new updates done or certificates revoked, no CRL present, or no CA certificate present.
- Number of sessions matching default SSL proxy profile in unified policy.
- Number of sessions dropped because of absence of default SSL proxy profile.

For details about the output fields of the command, see *show services ssl proxy counters*.

SSL Proxy Counters Information

IN THIS SECTION

Purpose

Display statistical counters for the SSL proxy session to provide information about the sessions.

Action

From the operational mode, use the **show services ssl proxy counters info** command.

```
user@host > show services ssl proxy counters info

Lsys Name : root-logical-system
PIC:fpc0 ------

sessions matched 0
sessions created 0
sessions destroyed 0
sessions ignored 0
sessions ignored : backup only 0
sessions whitelisted : IP based 0
sessions whitelisted : url based 0
crl : data added 1
crl : certificate revoked 0
crl : no crl info present 0
crl : no CA certificate 0
SSL sessions 0
SMTP over STARTTLS 0
```

```
IMAP over STARTTLS 0
POP3 over STARTTLS 0
SMTP sessions 0
IMAP sessions 0
POP3 sessions 0
Server not supporting STARTTLS 0
Client not supporting STARTTLS 0
Unified policy : default profile hit 0
Unified policy : no default profile 0
```

Meaning

The output display the counters details related SSL proxy session. These counters generally increment whenever there is some activity such as session matched, session created, and so on.

Example:

- Count of sessions created, matched, ignored or destroyed.
- Number of sessions allowlisted.
- Session counts based on CRL-related information such as new updates done, certificates revoked, no CRL present, or no CA certificate present.
- Number of sessions matching default SSL proxy profile in unified policy.
- Number of sessions dropped because of absence of default SSL proxy profile.

For details about the output fields of the command, see *show services ssl proxy counters*.

SSL Proxy Counters Errors

IN THIS SECTION

Purpose

Display statistical counters for the errors encountered in SSL proxy session.

Action

From the operational mode, use the **show services ssl proxy counters errors** command.

```
user@host > show services ssl proxy counters errors

Lsys Name : root-logical-system
PIC:fpc0 ------

Session create failed 0
non SSL sessions received 0
memory failures 0
session dropped 7
```

Meaning

The output display the counters details for the errors encountered in an SSL proxy session. Example:

- Number of failed sessions.
- Number of non-SSL sessions received on the system.
- Number of dropped sessions.

For details about the output fields of the command, see *show services ssl proxy counters*.

Display SSL Proxy Profile Details

IN THIS SECTION

Purpose

Display information about the SSL proxy profile.

Action

From the operational mode, use the **show services ssl proxy profile profile-name** command.

```
user@host > show services ssl proxy profile profile-name

Lsys Name : root-logical-system
PIC:fwdd0 fpc[0] pic[0] ------
Profile: ssl-proxy
enable-tracing: false
root-ca expired: false
allow non-ssl session: true
ssl-termination-id: 65537
ssl-initiation-id: 65537
Number of whitelist entries: 0
```

Meaning

Output of the command displays the details of the SSL proxy profile. Example:

- The number of sessions that are allowlisted.
- Whether the non SSL sessions are allowed.
- Whether the root certificate is active or expired.

For details about the output fields of the command, see *show services ssl proxy profile*.

Display SSL Proxy Profiles

IN THIS SECTION

Purpose

Display all the SSL proxy profiles configured on the device.

Action

From the operational mode, use the **show services ssl proxy profile all** command.

```
user@host > show services ssl proxy profile all

Lsys Name : root-logical-system
PIC:fwdd0 fpc[0] pic[0] ------
ID          Name
10         p1
11         p2
```

Meaning

The output displays the list of SSL proxy profiles available on the device.

For details about the output fields of the command, see *show services ssl proxy profile*.

Display SSL Proxy Session Cache Statistics

IN THIS SECTION

Purpose

Display the data for the SSL proxy session cache.

Action

From the operational mode, use the **show services ssl proxy session-cache statistics** command.

```
user@host > show services ssl proxy session-cache statistics

Lsys Name : root-logical-system
PIC: fpc0 fpc[0] pic[0]------------

Session cache hit                :          0
Session cache miss               :          0
Session cache full               :          0
```

Meaning

Command output displays SSL proxy session cache statistics. You can get the details such as number of times the information related to an SSL session is found in the cache or the number of times the information related to an SSL session is missing in the cache, and number of times the session cache limit is reached.

For details about the output fields of the command, see *show services ssl proxy session-cache statistics*.

Display SSL Proxy Session Cache Summary

IN THIS SECTION

Purpose

Display brief information about the entries stored in the SSL proxy session cache.

Action

From the operational mode, use the **show services ssl proxy session-cache entries summary** command.

```
user@host > show services ssl proxy session-cache entries summary

Lsys Name : root-logical-system
PIC:fwdd0 fpc[0] pic[0] ------
Hash Entry 1
Status: ACTIVE, Time to expire 294 seconds
Session Id Length: 32
Session Id: 1b 2a 9f 5f d8 6e d2 cd 6b b8 89 e8 88 07 75 80 32 c2 54 5a c7 9b 12 a2 e6 5c f0 6d
85 c5 40 4b
Dst IP: 5.0.0.1, Dst Port: 20753
SSL-T Profile Id: 2, SSL-I Profile Id: 2
```

Meaning

Command output displays SSL proxy session cache entries details such as session information saved in the cache, session status, session ID, and length of the session ID, destination IP address and port details, and SSL initiation and SSL termination profile IDs.

For details about the output fields of the command, see *show services ssl proxy session-cache entries*.

Display SSL Proxy Session Cache Details

IN THIS SECTION

Purpose

Display detail information about the entries stored in the SSL proxy session cache.

Action

From the operational mode, use the **show services ssl proxy session-cache entries detail** command.

```
user@host> show services ssl proxy session-cache entries detail
Lsys Name : root-logical-system
PIC: fpc0 fpc[0] pic[0
Hash Entry: 1
Status: ACTIVE, Time to expire 294 seconds
Session Id Length: 32
Session Id: c1 6e 88 65 43 9f 57 2f 0f 06 f7 4b 03 c5 38 58 74 b4 4f 43 66 9a 6f c7 a6 2a ae 22
ab f8 b4 ce
Dst IP: 5.0.0.1, Dst Port: 4433
SSL-T Profile Id: 2, SSL-I Profile Id: 2
Session Info:
Interdicted cert type [0x0]: CA issued, Authentication failed
Server cert verification result: unable to get local issuer certificate [0x14]
Server name extn len: 0, name: None
Server cert chain hash: b5 3d cd cb ca 35 81 5a db 6f 83 ab 5e a0 19 73

SSL-TERM session:
SSL ver: 0x303
Compression Method: 0
Cipher Id: 0x3000004
Master Key Length: 48

SSL-INIT session:
SSL ver: 0x303
Compression Method: 0
Cipher Id: 0x3000004
Master Key Length: 48

Hash Entry:2
Status: EXPIRED
Session Id Length: 32
Session Id: 1b 2a 9f 5f d8 6e d2 cd 6b b8 89 e8 88 07 75 80 32 c2 54 5a c7 9b 12 a2 e6 5c f0 6d
85 c5 40 4b
Dst IP: 5.0.0.1, Dst Port: 4433,
SSL-T Profile Id: 2, SSL-I Profile Id: 2
Session Info:
-------------
Interdicted cert type [0x0]: CA issued, Authentication failed
Server cert verification result: unable to get local issuer certificate [0x14]
Server name extn len: 0, name: None
Server cert chain hash: b5 3d cd cb ca 35 81 5a db 6f 83 ab 5e a0 19 73

SSL-TERM session:
----------------
SSL ver: 0x303
Compression Method: 0
Cipher Id: 0x3000004
Master Key Length: 48

SSL-INIT session:
----------------
SSL ver: 0x303
Compression Method: 0
Cipher Id: 0x3000004
Master Key Length: 48


Stale entry in cache: 1
```

Meaning

Command output displays cached SSL proxy session entries details. Example:

- Status of the cache entry with time to expire. Because the cache entries are valid only for short interval.
- Session ID, and length of the session ID.
- Destination IP address and destination port details.
- SSL initiation and SSL termination session details.
- Server certificate validation, interdicted certificate details.

For details about the output fields of the command, see *show services ssl proxy session-cache entries*.

Display SSL Proxy Certificate Cache Entry Statistics

IN THIS SECTION

Purpose

Display data for the SSL proxy certificate cache.

Action

From operational mode, use the **show services ssl proxy certificate–cache statistics** command.

```
user@host > show services ssl proxy certificate-cache statistics

Lsys Name : root-logical-system
PIC:fwdd0 fpc[0] pic[0] ------
cert cache hit 0
cert cache miss 0
cert cache full
```

Meaning

Command output displays SSL proxy certificate cache statistics such as number of times the match is available in cache, number of times an entry is not found in cache, or the number of times that cache was full.

For details about the output fields of the command, see *show services ssl proxy certificate?cache statistics*.

Display SSL Proxy Certificate Cache Entry Summary

IN THIS SECTION

Purpose

Display brief information about the entries stored in the SSL proxy certificate cache.

Action

From operational mode, use the **show services ssl proxy certificate-cache entries summary** command.

```
user@host > show services ssl proxy certificate-cache entries summary

Lsys Name : root-logical-system
PIC:fwdd0 fpc[0] pic[0] ------
Cache Entries : 1
Serial number : 0x12345678
SSL-I Profile Id: 1
Num of CRL updates: 0
```

Meaning

Command output displays certificate cache statistics such number of cache entries, serial number, profile ID, and CRL updates.

For details about the output fields of the command, see *show services ssl proxy certificate-cache entries*.

Display SSL Proxy Certificate Cache Entry Details

IN THIS SECTION

Purpose

Display detail information about the entries stored in the SSL proxy certificate cache.

Action

From operational mode, use the **show services ssl proxy certificate-cache entries detail** command.

```
user@host > show services ssl proxy certificate-cache entries detail

Lsys Name : root-logical-system
PIC:fwdd0 fpc[0] pic[0] ------
Cache entrie : 1
Serial number : 0x12345678
SSL-I Profile Id: 1
Num of CRL updates: 0
Status: Active: Time to expire 570 seconds

Cert Info:
-------------
Interdicted cert type [0x0]: CA issued, Authentication failed
Server cert verification result: unable to get local issuer certificate [0x14]
Cert reference count: 2
Subject: /C=IN/ST=KA/O=XYZ Inc/CN=ABC Inc Root CA/emailAddress=newca@test.com
Issuer: /CN=SSL-PROXY:DUMMY_CERT:GENERATED DUE TO SRVR AUTH FAILURE
```

Meaning

You can get the detail information about the cached SSL proxy certificate entries with this command. Example:

- Number of entries present in the certificate-cache.
- Number of times the CRL updates done till the interdicted certificate was added to the certificate-cache.
- Cached interdicted certificate and the server certificate verification results.
- Subject and issuer of the interdicted certificate.

For details about the output fields of the command, see *show services ssl proxy certificate-cache entries*.

Display SSL Proxy Status

IN THIS SECTION

Purpose

Display the status of the SSL proxy session.

Action

From operational mode, use the **show services ssl proxy status** command.

```
user@host > show services ssl proxy status
PIC:fwdd0 fpc[0] pic[0] ------
        One-Crypto       :  Enable
        Async Crypto     :  disable
```

```
    Proxy-activation :  Only if interested svcs configured
    Local Logging    :  disable
    SSLFP-PKID Link  :  UP
    Certificate cache : -
            Certificate Cache activated                : yes
            Invalidate certificate cache on CRL update : Disabled
            Max cert cache nodes  :       4000
            Cert cache node in use :          0
    Session cache : -
            Session cache activated : Activated
            Max session cache node  :      19660
            Session cache node in use     :          0
```

Meaning

The command displays the overall status of the SSL proxy. Example:

- Crypto status, proxy activation status.
- Certificate cache details such as whether certificate cache is activated, CRL configuration, certificate cache size, number of certificates in certificate cache currently used.
- Session cache details such as whether session cache is activated, size of the session cache, number of sessions in session cache currently used.

For details about the output fields of the command, see *show services ssl proxy status*.

Display SSL Termination Counter Details

IN THIS SECTION

Purpose

Display statistical counter details for the SSL termination sessions.

Action

From operational mode, use the **show services ssl termination counters all** command.

```
user@host > show services ssl termination counters all

Lsys Name : root-logical-system
PIC:fpc0 fpc[0] pic[0] ------

Memory errors                                  0
Handshake errors                               0
Cert Cache errors                              0
Server Protection errors                       0
Proxy errors                                   0
Crypto errors                                  0
Certificate errors                             0
One-Crypto errors                              0
Async-Crypto errors                            0
Mirror errors                                  0
handshakes started                             0
handshakes completed                           0
active sessions                                0
Interdicted cert generated                     0
proxy: sessions created                        0
proxy: sessions active                         0
proxy: sessions ignored                        0
proxy: renegotiation ignored                   0
proxy: session resumption                      0
proxy: secure renegotiation                    0
proxy: insecure renegotiation                  0
proxy: multiple renegotiation                  0
proxy: reneg after resumption                  0
init: passthrough requests                     0
init: start requests                           0
proxy: ECDSA based srvr auth                   0
proxy: RSA based srvr auth                     0
```

Meaning

You can get useful information about the SSL termination counters with this command. Example:

- Number of errors related to memory, handshake, certificate, server protection, proxy and crypto
- Number of sessions initiated handshake and completed handshake.
- Number of active sessions.
- Number of SSL proxy sessions such as sessions created, active sessions, ignored sessions, renegotiated sessions, sessions with different authentication methods and so on.

For details about the output fields of the command, see *show services ssl termination counters*.

Display SSL Termination Counters Errors

IN THIS SECTION

Purpose

Display statistical counters for the errors encountered in SSL termination session.

Action

From operational mode, use the **show services ssl termination counters error** command.

```
user@host > show services ssl termination counters error

Lsys Name : root-logical-system
PIC:fpc0 ------

Memory errors 0
```

```
Handshake errors 0
Cert Cache errors 0
Server Protection errors 0
Proxy errors 0
Crypto errors 0
Certificate errors 0
One-Crypto errors 0
Async-Crypto errors 0
Mirror errors 0
```

Meaning

The output of the command displays number of errors related to memory, handshake, certificate, server protection, proxy and crypto, and SSL decryption mirroring functionality.

For details about the output fields of the command, see *show services ssl termination counters*.

Display SSL Termination Counters Handshake

IN THIS SECTION

Purpose

Display statistical counters for the SSL termination handshake.

Action

From operational mode, use the **show services ssl termination counters handshake** command.

```
user@host > show services ssl termination counters handshake
```

```
Lsys Name : root-logical-system
PIC:fpc0 fpc[0] pic[0] ------

handshakes started 0
handshakes completed 0
active sessions 0
Interdicted cert generated 0
proxy: sessions created 0
proxy: sessions active 0
proxy: sessions ignored 0
proxy: renegotiation ignored 0
proxy: session resumption 0
proxy: secure renegotiation 0
proxy: insecure renegotiation 0
proxy: multiple renegotiation 0
proxy: reneg after resumption 0
init: passthrough requests 0
init: start requests 0
proxy: ECDSA based srvr auth 0
proxy: RSA based srvr auth 0
```

Meaning

You can get useful information about the SSL termination counters with this command. Example:

- Number of sessions initiated handshake and completed handshake.
- Number of active sessions
- Number of SSL proxy sessions such as sessions created, active sessions, ignored sessions, renegotiated sessions, sessions with different authentication methods and so on.

For details about the output fields of the command, see *show services ssl termination counters*.

Display SSL Termination Profile

IN THIS SECTION

Purpose

Display all SSL termination profiles available on the device.

Action

From operational mode, use the **show services ssl termination profile all** command.

```
user@host > show services ssl termination profile all
Lsys Name : root-logical-system
PIC:fwdd0 fpc[0] pic[0] ------
ID          Name
65536    p1_65536_proxy_t
65537    p2_65537_proxy_t
```

Meaning

The output of the command displays the list of all SSL termination profiles available on the device.

For details about the output fields of the command, see *show services ssl termination profile*.

Display SSL Termination Profile Summary

IN THIS SECTION

Purpose

Display the brief information about the SSL termination profiles.

Action

From operational mode, use the **show services ssl termination profile brief profile-name** command.

```
user@host > show services ssl termination profile brief profile-name

Lsys Name : root-logical-system

PIC: fwdd0 fpc[0] pic[0] ----------
Profile: ssl-termination
allow non-ssl session: true
preferred-ciphers: medium
Num of url categories configured: NIL
Number of whitelist entries: 0
```

Meaning

Displays the details of the SSL termination profile.

You can get useful information about the SSL initiation profile with this command. Example:

- Whether the root certificate is active or expired.
- Preferred SSL cipher with key strength.
- Whether the non SSL sessions are allowed.
- Number of URL categories configured.
- Number of allowlisted sessions.

For details about the output fields of the command, see *show services ssl termination profile*.

Display SSL Termination Profile Details

IN THIS SECTION

Purpose

Display the detail information about the SSL termination profile.

Action

From operational mode, use the **show services ssl termination profile detail profile-name** command.

```
user@host > show services ssl termination profile detail profile-name
Lsys Name : root-logical-system

PIC: fwdd0 fpc[0] pic[0] ----------

Profile                          : p1_65536_proxy_t
allow non-ssl session            : true
preferred-ciphers                : medium
Num of url categories configured : 0
Protocol version                 : all
Client Authentication            : notset
Server Authentication            : Required
Crypto Mode                      : hw-sync
Session Resumption               : Enabled
CRL check                        : Enabled
Certficate RSA : p_5
Renegotiation         : only secure allowed
Custom ciphers        : 0
Server cert           : 0
Decrypt Mirror        : Disabled
Trusted CA            : 0
```

```
        handshakes started            0
        handshakes completed          0
        active sessions               0
        total handshake errors        0
        Data Errors                   0
        session resumption            0
        secure renegotiation          0
        insecure renegotiation        0
        multiple renegotiation        0
        reneg after resumption        0
        no_reneg alert by peer        0
        drop on reneg                 0
```

Meaning

You can get useful information about the SSL termination profile with this command. Example:

- Profile name.
- Whether the non-SSL sessions are allowed.
- Category of the preferred cipher.
- Number of URL categories configured.
- Protocol version.
- Status of the various functionality such as client and server authentication, certificate revocation actions, session resumption, session renegotiation.
- Trusted CA and custom cipher details.
- SSL decryption mirror status.
- SSL termination per profile statistics or counters.

For details about the output fields of the command, see *show services ssl termination profile*.

Display SSL Initiation Counter Details

IN THIS SECTION

Purpose

Display statistical counters for the SSL initiation session.

Action

From operational mode, use the **show services ssl initiation counters all** command.

```
user@host > show services ssl initiation counters all
Lsys Name : root-logical-system

PIC:fpc0 fpc[0] pic[0] ------

Memory errors                                  0
Handshake errors                               0
Cert Cache errors                              0
Server Protection errors                       0
Proxy errors                                   0
Crypto errors                                  0
Certificate errors                             0
One-Crypto errors                              0
Async-Crypto errors                            0
Mirror errors                                  0
handshakes started                             0
handshakes completed                           0
active sessions                                0
Interdicted cert generated                     0
proxy: sessions created                        0
proxy: sessions active                         0
```

```
proxy: sessions ignored                        0
proxy: renegotiation ignored                   0
proxy: session resumption                      0
proxy: secure renegotiation                    0
proxy: insecure renegotiation                  0
proxy: multiple renegotiation                  0
proxy: reneg after resumption                  0
init: passthrough requests                     0
init: start requests                           0
proxy: ECDSA based srvr auth                   0
proxy: RSA based srvr auth                     0
```

Meaning

You can get useful information about the SSL initiation counters with this command. Example:

- Number of errors related to memory, handshake, certificate, server protection, proxy and crypto.
- Number of sessions initiated handshake and completed the handshake.
- Number of active sessions.
- Number of SSL proxy sessions such as sessions created, active sessions, ignored sessions, renegotiated sessions, sessions with different authentication methods and so on.

For details about the output fields of the command, see *show services ssl initiation counters*.

Display SSL initiation Counter Handshake

IN THIS SECTION

Purpose

Display statistical counters for the SSL initiation handshake.

Action

From operational mode, use the **show services ssl initiation counters handshake** command.

```
user@host > show services ssl initiation counters handshake
Lsys Name : root-logical-system

PIC:fpc0 fpc[0] pic[0] ------

handshakes started 0
handshakes completed 0
active sessions 0
Interdicted cert generated 0
proxy: sessions created 0
proxy: sessions active 0
proxy: sessions ignored 0
proxy: renegotiation ignored 0
proxy: session resumption 0
proxy: secure renegotiation 0
proxy: insecure renegotiation 0
proxy: multiple renegotiation 0
proxy: reneg after resumption 0
init: passthrough requests 0
init: start requests 0
proxy: ECDSA based srvr auth 0
proxy: RSA based srvr auth 0
```

Meaning

You can get useful information about the SSL initiation counters with this command. Example:

- Number of sessions initiated handshake and completed handshake.
- Number of active sessions.
- Number of SSL proxy sessions such as sessions created, active sessions, ignored sessions, renegotiated sessions, sessions with different authentication methods and so on.

For details about the output fields of the command, see *show services ssl initiation counters*.

Display SSL Initiation Counter Errors

IN THIS SECTION

Purpose

Display statistical counters for the errors encountered in SSL initiation session.

Action

From operational mode, use the **show services ssl initiation counters error** command.

```
user@host > show services ssl initiation counters error
Lsys Name : root-logical-system

PIC:fpc0 fpc[0] pic[0] ------

Memory errors 0
Handshake errors 0
Cert Cache errors 0
Server Protection errors 0
Proxy errors 0
Crypto errors 0
Certificate errors 0
One-Crypto errors 0
Async-Crypto errors 0
Mirror errors 0
```

Meaning

The output of the command displays number of errors related to memory, handshake, certificate, server protection, proxy and crypto, and SSL decryption mirroring functionality.

For details about the output fields of the command, see *show services ssl initiation counters*.

Display SSL Initiation Profile

IN THIS SECTION

Purpose

Display all SSL initiation profiles available on the device.

Action

From operational mode, use the **show services ssl initiation profile all** command.

```
user@host > show services ssl initiation profile all
```

```


Lsys Name : root-logical-system

PIC: fwdd0 fpc[0] pic[0] ----------

ID          Name

65536  SSL_PROFILE_65536_proxy_i
```

Meaning

The output of the command displays the list of all SSL initiation profiles available on the device.

For details about the output fields of the command, see *show services ssl initiation profile*.

Display SSL Initiation Profile Summary

IN THIS SECTION

Purpose

Display the summary of the SSL initiation profile.

Action

From operational mode, use the **show services ssl initiation profile brief profile-name** command.

```
user@host > show services ssl initiation profile brief profile-name
Lsys Name : root-logical-system

PIC: fpc0 fpc[0] pic[0] ----------

Profile                          : SSL_PROFILE_65536_proxy_i
allow non-ssl session            : true
preferred-ciphers                : medium
Num of url categories configured : 0
```

Meaning

Displays the details of the SSL initiation profile such as profile name, whether the non-SSL sessions ar allowed, prefered-ciphers, and number of URL categories configured.

For details about the output fields of the command, see *show services ssl initiation profile*.

Display SSL Initiation Profile Details

IN THIS SECTION

Purpose

Display the detail information about the SSL initiation profile.

Action

From operational mode, use the **show services ssl initiation profile detail profile-name** command.

```
user@host > show services ssl initiation profile detail profile-name
Lsys Name : root-logical-system

PIC: fpc0 fpc[0] pic[0] ----------

Profile                          : SSL_PROFILE_65536_proxy_i
allow non-ssl session            : true
preferred-ciphers                : medium
Num of url categories configured : 0
Protocol version                 : all
Client Authentication            : notset
Server Authentication            : Ignore Failure
Crypto Mode                      : sw
```

```
Session Resumption               : Enabled
CRL check                        : Enabled
Certficate RSA : ssl-inspect-ca
Renegotiation        : only secure allowed
Custom ciphers       : 0
Server cert          : 0
Decrypt Mirror       : Disabled
Trusted CA           : 1
       handshakes started            8
       handshakes completed          8
       active sessions               0
       total handshake errors        0
       Data Errors                   0
       session resumption            5
       secure renegotiation          0
       insecure renegotiation        0
       multiple renegotiation        0
       reneg after resumption        0
       no_reneg alert by peer        0
       drop on reneg                 0
```

Meaning

You can get useful information about the SSL initiation profile with this command. Example:

- Whether the non SSL sessions are allowed.
- Preferred SSL cipher
- Number of URL categories configured.
- Status of the various functionality such as client and server authentication, certificate revocation actions, session resumption, session renegotiation.
- Trusted CA, chain certificates details.
- SSL decryption mirror status
- SSL initiation session counters

For details about the output fields of the command, see *show services ssl initiation profile*.

Display SSL Drop Log Details

IN THIS SECTION

Purpose

Display information about SSL drop logs.

Action

From operational mode, use the **show services ssl droplogs** command.

```
user@host > show services ssl droplogs

Lsys Name : root-logical-system
PIC:fpc0 fpc[0] pic[0]-------

 ===========log mesg for cpu 0

 ===========log mesg for cpu 1

 log mesg is  File: ../../../../../../../../../src/junos/jsf/plugin/ssl/jssl_common.c Function:
jssl_X509_verify_cert Line: 3767 Message: unable to get local issuer certificate C2S plugin
chain : [Plugin junos-jdpi: action: ignore]-> [Plugin junos-tcp-svr-emul: action: none]->
[Plugin junos-ssl-proxy: action: ignore]-> [Plugin junos-ssl-term: action: none]-> [Plugin junos-
dpi-stream: action: none]-> [Plugin junos-idp-stream: action: ignore]-> [Plugin junos-ssl-init:
action: none]-> [Plugin junos-tcp-clt-emul: action: none] S2C plugin chain: [Plugin junos-jdpi:
action: ignore]-> [Plugin junos-tcp-clt-emul: action: none]-> [Plugin junos-ssl-init: action:
none]-> [Plugin junos-dpi-stream: action: none]-> [Plugin junos-idp-stream: action: ignore]->
[Plugin junos-ssl-term: action: none]-> [Plugin junos-ssl-proxy: action: ignore]-> [Plugin junos-
tcp-svr-emul: action: none] SourceIP:5.0.0.1 DestIP:4.0.0.1 Source Port:40281 Dest Port:443
```

```
source interface:ge-0/0/1.0 Destination interface:ge-0/0/0.0 source zone:untrust destination
Zone:trust
```

Meaning

Output of the command displays the denied/dropped session details. You can use the command output to understand the issue why session was dropped.

5

CHAPTER

Configuration Statements and Operational Commands

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Learn about the syntax and options that make up the statements and commands and 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*
- *CLI Commands*