JUNIPER NETWORKS

Junos® OS

Common Criteria and FIPS Evaluated Configuration Guide for SRX4600 Devices

Release
18.1R1

Modified: 2019-01-02

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

Junos® OS Common Criteria and FIPS Evaluated Configuration Guide for SRX4600 Devices
18.1R1

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

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

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

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

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

Documentation Conventions

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

Table 1: Notice Icons

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

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

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*

Table 2: Text and Syntax Conventions (continued)

Convention	Description	Examples
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

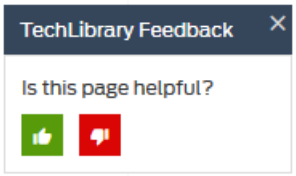
Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

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

Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. You can use either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the Juniper Networks TechLibrary site, and do one of the following:



  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide feedback.

- E-mail—Send your comments to **techpubs-comments@juniper.net**. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

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

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

Self-Help Online Tools and Resources

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

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

- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Create a service request online: https://myjuniper.juniper.net

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

Creating a Service Request with JTAC

You can create a service request with JTAC on the Web or by telephone.

- Visit https://myjuniper.juniper.net.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see https://support.juniper.net/support/requesting-support/.

CHAPTER 1

Overview

Understanding the Common Criteria Evaluated Configuration

This document describes the steps required to duplicate the configuration of the device running Junos OS when the device is evaluated. This is referred to as the evaluated configuration. The following list describes the standards to which the device has been evaluated:

- Collaborative Protection Profile for Network Devices, version 2.0, 4 May 2017 (NDcPP)
- Collaborative Protection Profile for Stateful Traffic Filter Firewalls, Version 1.0, 27 February 2015 (FWcPP)
- Collaborative Protection Profile for Network Devices or Collaborative Protection Profile for Stateful Traffic Filter Firewalls Extended Package (EP) for Intrusion Prevention Systems (IPS), Version 2.11, 15 June 2017 (IPSEP)
- Network Device Collaborative Protection Profile (NDcPP)/Stateful Traffic Filter Firewall Collaborative Protection Profile (FWcPP) Extended Package VPN Gateway, Version 2.1, 8 March 2017 (VPNEP)

These documents are available at https://www.niap-ccevs.org/Profile/PP.cfm?archived=1.

NOTE: The Junos certified version is Junos version 18.1R1. On SRX4600 devices, Junos OS Release 18.1R1 is certified for Common Criteria with FIPS mode enabled on the devices.

Understanding Common Criteria

Common Criteria for information technology is an international agreement signed by 28 countries that permits the evaluation of security products against a common set of standards. In the Common Criteria Recognition Arrangement (CCRA) at http://www.commoncriteriaportal.org/ccra/, the participants agree to mutually recognize evaluations of products performed in other countries. All evaluations are performed using a common methodology for information technology security evaluation.

For more information on Common Criteria, see http://www.commoncriteriaportal.org/.

Supported Platforms

The SRX4600 devices support the IPSEP, NDPP, FWEP, and VPNEP features described in this document.

Related Documentation

- Identifying Secure Product Delivery on page 22

Understanding Junos OS in FIPS Mode of Operation

Federal Information Processing Standards (FIPS) 140-2 defines security levels for hardware and software that perform cryptographic functions. Junos-FIPS is a version of the Junos operating system (Junos OS) that complies with Federal Information Processing Standard (FIPS) 140-2.

Operating SRX Series devices in a FIPS 140-2 Level 2 environment requires enabling and configuring FIPS mode of operation on the device from the Junos OS command-line interface (CLI).

The *Cryptographic Officer* enables FIPS mode of operation in Junos OS Release 18.1R1 and sets up keys and passwords for the system and other *FIPS users* who can view the configuration. Both user types can also perform normal configuration tasks on the device (such as modify interface types) as individual user configuration allows.

BEST PRACTICE: **Be sure to verify the secure delivery of your device and apply tamper-evident seals to its vulnerable ports.**

- About the Cryptographic Boundary on Your Device on page 18
- How FIPS Mode of Operation Differs from Non-FIPS Mode of Operation on page 19
- Validated Version of Junos OS in FIPS Mode of Operation on page 19

About the Cryptographic Boundary on Your Device

FIPS 140-2 compliance requires a defined *cryptographic boundary* around each *cryptographic module* on a device. Junos OS in FIPS mode of operation prevents the cryptographic module from running any software that is not part of the FIPS-certified distribution, and allows only FIPS-approved cryptographic algorithms to be used. No

critical security parameters (CSPs), such as passwords and keys, can cross the cryptographic boundary of the module by, for example, being displayed on a console or written to an external log file.

CAUTION: **Virtual Chassis features are not supported in FIPS mode of operation. Do not configure a Virtual Chassis in FIPS mode of operation.**

To physically secure the cryptographic module, all Juniper Networks devices require a tamper-evident seal on the USB and mini-USB ports.

How FIPS Mode of Operation Differs from Non-FIPS Mode of Operation

Unlike Junos OS in non-FIPS mode of operation, Junos OS in FIPS mode of operation is a *nonmodifiable operational environment*. In addition, Junos OS in FIPS mode of operation differs in the following ways from Junos OS in non-FIPS mode of operation:

- Self-tests of all cryptographic algorithms are performed at startup.
- Self-tests of random number and key generation are performed continuously.
- Weak cryptographic algorithms such as Data Encryption Standard (DES) and MD5 are disabled.
- Weak or unencrypted management connections must not be configured.
- Passwords must be encrypted with strong one-way algorithms that do not permit decryption.
- Junos-FIPS administrator passwords must be at least 10 characters long.
- Cryptographic keys must be encrypted before transmission.

The FIPS 140-2 standard is available for download from the National Institute of Standards and Technology (NIST) at: http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

Validated Version of Junos OS in FIPS Mode of Operation

To determine the validated version of Junos OS in FIPS mode of operation and for regulatory compliance information about Common Criteria and FIPS for Juniper Networks products, see the Juniper Networks Compliance Advisor.

Related Documentation

- Identifying Secure Product Delivery on page 22

Understanding FIPS Mode of Operation Terminology and Supported Cryptographic Algorithms

Use the definitions of FIPS terms and supported algorithms to help you understand Junos OS in FIPS mode of operation.

- FIPS Terminology on page 20
- Supported Cryptographic Algorithms on page 21

FIPS Terminology

Critical security parameter (CSP)—Security-related information—for example, secret and private cryptographic keys and authentication data such as passwords and personal identification numbers (PINs)—whose disclosure or modification can compromise the security of a cryptographic module or the information it protects.

Cryptographic module—The set of hardware, software, and firmware that implements approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary. SRX Series devices are certified at FIPS 140-2 Level 2.

Cryptographic Officer—Person with appropriate permissions who is responsible for securely enabling, configuring, monitoring, and maintaining Junos OS in FIPS mode of operation on a device. For details, see "Understanding Roles and Services for Junos OS in FIPS Mode of Operation" on page 29.

ESP—Encapsulating Security Payload (ESP) protocol. The part of the IPsec protocol that guarantees the confidentiality of packets through encryption. The protocol ensures that if an ESP packet is successfully decrypted, and no other party knows the secret key the peers share, the packet was not wiretapped in transit.

FIPS—Federal Information Processing Standards. FIPS 140-2 specifies requirements for security and cryptographic modules. Junos OS in FIPS mode of operation complies with FIPS 140-2 Level 2.

IKE—The Internet Key Exchange (IKE) is part of IPsec and provides ways to securely negotiate the shared private keys that the authentication header (AH) and ESP portions of IPsec need to function properly. IKE employs Diffie-Hellman key-exchange methods and is optional in IPsec. (The shared keys can be entered manually at the endpoints.)

IPsec—The IP Security (IPsec) protocol. A standard way to add security to Internet communications. An IPsec security association (SA) establishes secure communication with another FIPS cryptographic module by means of mutual authentication and encryption.

KATs—Known answer tests. System self-tests that validate the output of cryptographic algorithms approved for FIPS and test the integrity of some Junos OS modules. For details, see "Understanding FIPS Self-Tests" on page 129.

SA—Security association (SA). A connection between hosts that allows them to communicate securely by defining, for example, how they exchange private keys. As Cryptographic Officer, you must manually configure an internal SA on devices running Junos OS in FIPS mode of operation. All values, including the keys, must be statically specified in the configuration.

SPI—Security parameter index (SPI). A numeric identifier used with the destination address and security protocol in IPsec to identify an SA. Because you manually configure the SA for Junos OS in FIPS mode of operation, the SPI must be entered as a parameter rather than derived randomly.

SSH—A protocol that uses strong authentication and encryption for remote access across a nonsecure network. SSH provides remote login, remote program execution, file copy, and other functions. It is intended as a secure replacement for **rlogin**, **rsh**, and **rcp** in a UNIX environment. To secure the information sent over administrative connections, use SSHv2 for CLI configuration. In Junos OS, SSHv2 is enabled by default, and SSHv1, which is not considered secure, is disabled.

Zeroization—Erasure of all CSPs and other user-created data on a device before its operation as a FIPS cryptographic module—or in preparation for repurposing the device for non-FIPS operation. The Cryptographic Officer can zeroize the system with a CLI operational command. For details, see "Understanding Zeroization to Clear System Data for FIPS Mode of Operation" on page 35.

RSA—RSA is a public-key cryptosystems which is widely used for secure data transmission. In the cryptosystem, the encryption key is public and the decryption key is private. A user can create and publish a public key based on two large prime numbers, along with an auxiliary value. The prime numbers must be kept private.

Supported Cryptographic Algorithms

Each implementation of an algorithm is checked by a series of known answer test (KAT) self-tests. Any self-test failure results in a FIPS error state.

BEST PRACTICE: **For FIPS 140-2 compliance, use only FIPS-approved cryptographic algorithms in Junos OS in FIPS mode of operation.**

The following cryptographic algorithms are supported in FIPS mode of operation. Symmetric methods use the same key for encryption and decryption, while asymmetric methods (preferred) use different keys for encryption and decryption.

AES—The Advanced Encryption Standard (AES), defined in FIPS PUB 197. The AES algorithm uses keys of 128, 192, or 256 bits to encrypt and decrypt data in blocks of 128 bits.

Diffie-Hellman—A method of key exchange across a nonsecure environment (such as the Internet). The Diffie-Hellman algorithm negotiates a session key without sending the key itself across the network by allowing each party to pick a partial key independently and send part of that key to the other. Each side then calculates a common key value. This is a symmetrical method, and keys are typically used only for a short time, discarded, and regenerated.

ECDH—Elliptic Curve Diffie-Hellman. A variant of the Diffie-Hellman key exchange algorithm that uses cryptography based on the algebraic structure of elliptic curves over finite fields. ECDH allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel. The shared secret can be used either as a key or to derive another key for encrypting subsequent communications using a symmetric key cipher.

ECDSA—Elliptic Curve Digital Signature Algorithm. A variant of the Digital Signature Algorithm (DSA) that uses cryptography based on the algebraic structure of elliptic

curves over finite fields. The bit size of the elliptic curve determines the difficulty of decrypting the key. The public key believed to be needed for ECDSA is about twice the size of the security level, in bits. ECDSA using the P-256, P-384, or the P-521 curve can be configured under OpenSSH.

HMAC—Defined as "Keyed-Hashing for Message Authentication" in RFC 2104, HMAC combines hashing algorithms with cryptographic keys for message authentication. For Junos OS in FIPS mode of operation, HMAC uses the iterated cryptographic hash function SHA-1 (designated as HMAC-SHA1) along with a secret key.

3DES (3des-cbc)—Encryption standard based on the original Data Encryption Standard (DES) from the 1970s that used a 56-bit key and was cracked in 1997. The more secure 3DES is DES enhanced with three multiple stages and effective key lengths of about 112 bits. For Junos OS in FIPS mode of operation, 3DES is implemented with cipher block chaining (CBC).

Related Documentation

- Understanding Zeroization to Clear System Data for FIPS Mode of Operation on page 35
- Understanding FIPS Self-Tests on page 129

Identifying Secure Product Delivery

There are several mechanisms provided in the delivery process to ensure that a customer receives a product that has not been tampered with. The customer should perform the following checks upon receipt of a device to verify the integrity of the platform.

- Shipping label—Ensure that the shipping label correctly identifies the correct customer name and address as well as the device.
- Outside packaging—Inspect the outside shipping box and tape. Ensure that the shipping tape has not been cut or otherwise compromised. Ensure that the box has not been cut or damaged to allow access to the device.
- Inside packaging—Inspect the plastic bag and seal. Ensure that the bag is not cut or removed. Ensure that the seal remains intact.

If the customer identifies a problem during the inspection, he or she should immediately contact the supplier. Provide the order number, tracking number, and a description of the identified problem to the supplier.

Additionally, there are several checks that can be performed to ensure that the customer has received a box sent by Juniper Networks and not a different company masquerading as Juniper Networks. The customer should perform the following checks upon receipt of a device to verify the authenticity of the device:

- Verify that the device was ordered using a purchase order. Juniper Networks devices are never shipped without a purchase order.
- When a device is shipped, a shipment notification is sent to the e-mail address provided by the customer when the order is taken. Verify that this e-mail notification was received. Verify that the e-mail contains the following information:

- Purchase order number
- Juniper Networks order number used to track the shipment
- Carrier tracking number used to track the shipment
- List of items shipped including serial numbers
- Address and contacts of both the supplier and the customer

- Verify that the shipment was initiated by Juniper Networks. To verify that a shipment was initiated by Juniper Networks, you should perform the following tasks:
  - Compare the carrier tracking number of the Juniper Networks order number listed in the Juniper Networks shipping notification with the tracking number on the package received.
  - Log on to the Juniper Networks online customer support portal to view the order status. Compare the carrier tracking number or the Juniper Networks order number listed in the Juniper Networks shipment notification with the tracking number on the package received.

Related Documentation

- Understanding the Common Criteria Evaluated Configuration on page 17

Understanding Management Interfaces

The following management interfaces can be used in the evaluated configuration:

- Local Management Interfaces—The RJ-45 console port on the rear panel of a device is configured as RS-232 data terminal equipment (DTE). You can use the command-line interface (CLI) over this port to configure the device from a terminal.
- Remote Management Protocols—The device can be remotely managed over any Ethernet interface. SSHv2 is the only permitted remote management protocol that can be used in the evaluated configuration, and it is enabled by default on the device. The remote management protocols J-Web and Telnet are not available for use on the device.

Related Documentation

- Understanding the Common Criteria Evaluated Configuration on page 17

CHAPTER 2

Configuring Administrative Credentials and Privileges

Understanding the Associated Password Rules for an Authorized Administrator

The authorized administrator is associated with a defined login class, and the administrator is assigned with all permissions. Data is stored locally for fixed password authentication.

NOTE: We recommend that you not use control characters in passwords.

Use the following guidelines and configuration options for passwords and when selecting passwords for authorized administrator accounts. Passwords should be:

- Easy to remember so that users are not tempted to write it down.
- Changed periodically.
- Private and not shared with anyone.
- Contain a minimum of 10 characters. The minimum password length is 10 characters.

```
[ edit ]
administrator@host# set system login password minimum-length 10
```

- Include both alphanumeric and punctuation characters, composed of any combination of upper and lowercase letters, numbers, and special characters such as, “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”. There should be at least a change in one case, one or more digits, and one or more punctuation marks.
- Contain character sets. Valid character sets include uppercase letters, lowercase letters, numbers, punctuation, and other special characters.

```
[ edit ]
administrator@host# set system login password change-type character-sets
```

- Contain the minimum number of character sets or character set changes. The minimum number of character sets required in plain-text passwords in Junos FIPS is 2.

```
[ edit ]
administrator@host# set system login password minimum-changes 2
```

NOTE: The authentication algorithm for plain-text passwords must be configured as sha256.

```
[ edit ]
administrator@host# set system login password format sha256
```

Weak passwords are:

- Words that might be found in or exist as a permuted form in a system file such as **/etc/passwd**.
- The hostname of the system (always a first guess).
- Any words appearing in a dictionary. This includes dictionaries other than English, and words found in works such as Shakespeare, Lewis Carroll, Roget's Thesaurus, and so on. This prohibition includes common words and phrases from sports, sayings, movies, and television shows.
- Permutations on any of the above. For example, a dictionary word with vowels replaced with digits (for example f00t) or with digits added to the end.
- Any machine-generated passwords. Algorithms reduce the search space of password-guessing programs and so should not be used.

Strong reusable passwords can be based on letters from a favorite phrase or word, and then concatenated with other, unrelated words, along with additional digits and punctuation.

NOTE: Passwords should be changed periodically.

Configuring a Network Device Protection Profile Authorized Administrator

An account for **root** is always present in a configuration and is not intended for use in normal operation. In the evaluated configuration, the **root** account is restricted to the initial installation and configuration of the evaluated device.

An NDPP authorized administrator must have all permissions, including the ability to change the router configuration.

To configure an authorized administrator:

1. Create a login class named security-admin with all permissions.

```
[edit]
root@host# set system login class security-admin permissions all
```

2. Define your NDPP user authorized administrator.

```
[edit]
root@host# set system login user NDPP-user full-name Common Criteria NDPP
   Authorized Administrator class security-admin authentication encrypted-password
   <password>
```

3. Configure the authentication algorithm for plain-text passwords as sha256.

```
[edit]
root@host# set system login password format sha256
```

4. Commit the changes.

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

NOTE: The root password should be reset following the change to sha256 for the password storage format. This ensures the new password is protected using a sha256 hash, rather than the default password hashing algorithm. To reset the root password, use the set system login user root password *password* command, and confirm the new password when prompted.

Related Documentation

- Understanding the Associated Password Rules for an Authorized Administrator on page 25

CHAPTER 3

Configuring Roles and Authentication Methods

Understanding Roles and Services for Junos OS in FIPS Mode of Operation

The Juniper Networks Junos operating system (Junos OS) running in non-FIPS mode of operation allows a wide range of capabilities for users, and authentication is identity-based. In contrast, the FIPS 140-2 standard defines two user roles: *Cryptographic Officer* and *FIPS user*. These roles are defined in terms of Junos OS user capabilities.

All other user types defined for Junos OS in FIPS mode of operation (operator, administrative user, and so on) must fall into one of the two categories: Cryptographic Officer or FIPS user. For this reason, user authentication in FIPS mode of operation is role-based rather than identity-based.

In addition to their FIPS roles, both user types can perform normal configuration tasks on the device as individual user configuration allows.

Cryptographic Officers and FIPS users perform all FIPS-related configuration tasks and issue all statements and commands for Junos OS in FIPS mode of operation. Cryptographic Officer and FIPS user configurations must follow the guidelines for Junos OS in FIPS mode of operation.

For details, see:

- Cryptographic Officer Role and Responsibilities on page 30
- FIPS User Role and Responsibilities on page 30
- What Is Expected of All FIPS Users on page 31

Cryptographic Officer Role and Responsibilities

The Cryptographic Officer is the person responsible for enabling, configuring, monitoring, and maintaining Junos OS in FIPS mode of operation on a device. The Cryptographic Officer securely installs Junos OS on the device, enables FIPS mode of operation, establishes keys and passwords for other users and software modules, and initializes the device before network connection. The Cryptographic Officer can configure and monitor the module through a console or SSH connection.

BEST PRACTICE: We recommend that the Cryptographic Officer administer the system in a secure manner by keeping passwords secure and checking audit files.

The permissions that distinguish the Cryptographic Officer from other FIPS users are **secret**, **security**, **maintenance**, and **control**. For FIPS compliance, assign the Cryptographic Officer to a login class that contains all of these permissions. A user with the Junos OS maintenance permission can read files containing critical security parameters (CSPs).

NOTE: Junos OS in FIPS mode of operation does not support the *FIPS 140-2 maintenance role*, which is different from the Junos OS maintenance permission.

Among the tasks related to Junos OS in FIPS mode of operation, the Cryptographic Officer is expected to:

- Set the initial root password.
- Reset user passwords for FIPS-approved algorithms during upgrades from Junos OS.
- Set up manual IPsec SAs for configuration with dual Routing Engines.
- Examine log and audit files for events of interest.
- Erase user-generated files and data on (zeroize) the device.

FIPS User Role and Responsibilities

All FIPS users, including the Cryptographic Officer, can view the configuration. Only the user assigned as the Cryptographic Officer can modify the configuration.

The permissions that distinguish Cryptographic Officers from other FIPS users are **secret**, **security**, **maintenance**, and **control**. For FIPS compliance, assign the FIPS user to a class that contains *none* of these permissions.

FIPS users configure networking features on the device and perform other tasks that are not specific to FIPS mode of operation. FIPS users who are not Cryptographic Officers can perform reboots and view status output.

What Is Expected of All FIPS Users

All FIPS users, including the Cryptographic Officer, must observe security guidelines at all times.

All FIPS users must:

- Keep all passwords confidential.
- Store devices and documentation in a secure area.
- Deploy devices in secure areas.
- Check audit files periodically.
- Conform to all other FIPS 140-2 security rules.
- Follow these guidelines:
  - Users are trusted.
  - Users abide by all security guidelines.
  - Users do not deliberately compromise security.
  - Users behave responsibly at all times.

Related Documentation

- Understanding FIPS Mode of Operation Terminology and Supported Cryptographic Algorithms on page 19
- Understanding Zeroization to Clear System Data for FIPS Mode of Operation on page 35

Understanding Services for Junos OS in FIPS Mode of Operation

All services implemented by the module are listed in the tables that follow.

Understanding Authenticated Services

Table 3 on page 31lists the authenticated services on the device running Junos OS.

Table 3: Authenticated services

Authenticated Services	Description	Cryptographic Officer	User (read-only)	User (network)
Configure security	Security relevant configuration	x	–	–
Configure	Non-security relevant configuration	x	–	–
Secure traffic	IPsec protected routing	–	–	x

Table 3: Authenticated services (continued)

Authenticated Services	Description	Cryptographic Officer	User (read-only)	User (network)
Status	Display the status	x	x	–
Zeroize	Destroy all critical security parameters (CSPs)	x	–	–
SSH connect	Initiate SSH connection for SSH monitoring and control (CLI)	x	x	–
IPsec connect	Initiate IPsec connection (IKE)	x	–	x
Console access	Console monitoring and control (CLI)	x	x	–
Remote reset	Software-initiated reset	x	–	–

Table 4: Unauthenticated traffic

Service	Description
Local reset	Hardware reset or power cycle
Traffic	Traffic requiring no cryptographic services

Critical Security Parameters

Critical security parameters (CSPs) are security-related information such as cryptographic keys and passwords that can compromise the security of the cryptographic module or the security of the information protected by the module if they are disclosed or modified.

Zeroization of the system erases all traces of CSPs in preparation for operating the device as a cryptographic module.

Table 5 on page 32 lists the CSP access rights within services.

Table 5: CSP Access Rights Within Services

Service	CSPs					
	DRBG_Seed	DRBG_State	SSH PHK	SSH DH	SSH-SEK	ESP-SEK
Configure security	–	E	G, W	–	–	–
Configure	–	–	–	–	–	–
Secure Traffic	–	–	–	–	–	E
Status	–	–	–	–	–	–
Zeroize	Z	Z	Z	Z	Z	Z

Table 5: CSP Access Rights Within Services (continued)

Service	CSPs					
	DRBG_Seed	DRBG_State	SSH PHK	SSH DH	SSH-SEK	ESP-SEK
SSH connect	–	E	E	G, E	G, E	–
IPSec connect	–	E	–	–	–	G
Console access	–	–	–	–	–	–
Remote reset	G, E	G	–	Z	Z	Z
Local Reset	G, E	G	–	Z	Z	Z
Traffic	–	–	–	–	–	–

Service	CSPs				
	IKE-PSK	IKE-Priv	IKE-SKEYI	IKE-SKE	IKE-DH-PRI
Configure security	W	G, W	–	–	–
Configure	–	–	–	–	–
Secure Traffic	–	–	–	E	–
Status	–	–	–	–	–
Zeroize	Z	Z	–	–	–
SSH connect	–	–	–	–	–
IPSec connect	E	E	G	G	G
Console access	–	–	–	–	–
Remote reset	–	–	Z	Z	Z
Local Reset	–	–	Z	Z	Z
Traffic	–	–	–	–	–

Here:

- G = Generate: The device generates the CSP.
- E = Execute: The device runs using the CSP.

- W = Write: The CSP is updated or written to the device.
- Z = Zeroize: The device zeroizes the CSP.

Related Documentation

- Understanding Zeroization to Clear System Data for FIPS Mode of Operation on page 35
- Understanding FIPS Authentication Methods on page 43

Downloading Software Packages from Juniper Networks (FIPS Mode)

You can download the following Junos OS software packages from the Juniper Networks website:

- Junos OS for SRX4600, Release 18.1R1
- Junos FIPS mode, Release 18.1R1

Before you begin to download the software, ensure that you have a Juniper Networks Web account and a valid support contract. To obtain an account, complete the registration form at the Juniper Networks website: https://www.juniper.net/registration/Register.jsp.

To download software packages from Juniper Networks:

1. Using a Web browser, follow the links to the download URL on the Juniper Networks webpage.

   https://www.juniper.net/support/downloads/junos.html

2. Log in to the Juniper Networks authentication system using the username (generally your e-mail address) and password supplied by Juniper Networks representatives.

3. Download the software. See Downloading Software

Related Documentation

- Installation and Upgrade Guide

Installing Junos Software Packages (FIPS Mode)

SRX Series devices can provide the security defined by Federal Information Processing Standards (FIPS) 140-2 Level 2 if these devices are operated in the Junos OS in FIPS mode. To operate in Junos OS in FIPS mode, the device must have the following software packages installed:

- Junos OS for SRX4600, Release 18.1R1
- Junos FIPS mode, Release 18.1R1

NOTE: Junos OS is delivered in signed packages that contain digital signatures to ensure the Juniper Networks software is running. When installing the software packages, Junos OS validates the signatures and the public key certificates used to digitally sign the software packages. If the signature or certificate is found to be invalid (for example, when the certificate validity period has expired or cannot be verified against the root CA stored in the Junos OS internal store), the installation process fails.

To install these software packages, perform the following tasks:

1. Download the Junos OS package and the Junos FIPS mode package from https://www.juniper.net/. See Downloading Software.

2. Install the Junos OS on your device using a TFTP server, see Installing Junos OS on SRX Series Devices from the Boot Loader Using a TFTP Server or install Junos OS on your device using the following CLI command: **request system software add /<image-path>/<junos package> no-copy no-validate reboot**.

Related Documentation

- Installation and Upgrade Guide

Understanding Zeroization to Clear System Data for FIPS Mode of Operation

Zeroization completely erases all configuration information on the device, including all plaintext passwords, secrets, and private keys for SSH, local encryption, local authentication, and IPsec. To exit the FIPS mode you need to zeroize the device.

The cryptographic module provides a non-approved mode of operation in which non-approved cryptographic algorithms are supported. When moving from the non-approved mode of operation to the approved mode of operation, the Cryptographic Officer must zeroize the non-approved mode critical security parameters (CSPs). The Cryptographic Officer initiates the zeroization process by entering the **request system zeroize hypervisor** operational command from the CLI after enabling FIPS mode of operation. Use of this command is restricted to the Cryptographic Officer.

CAUTION: Perform system zeroization with care. After the zeroization process is complete, no data is left on the device. This command erases all the CSPs, configurations, and the hard disk partitions containing the device image. Hence, the device does not boot up on zeroization and USB reimage is required to recover the device.

Zeroization can be time-consuming. Although all configurations are removed in a few seconds, the zeroization process goes on to overwrite all media, which can take considerable time depending on the size of the media.

- Why Zeroize? on page 36
- When to Zeroize? on page 36

Why Zeroize?

Your device is not considered a valid FIPS cryptographic module until all CSPs have been entered—or reentered—while the device is in FIPS mode of operation.

BEST PRACTICE: **For FIPS 140-2 compliance, we recommend that you zeroize the device to exit the FIPS mode.**

When to Zeroize?

As a Cryptographic Officer, perform zeroization in the following situations:

- **Before FIPS operation**—To prepare your device for operation as a FIPS cryptographic module, perform zeroization to remove the non-approved mode critical security parameters (CSPs) and enable FIPS mode on the device.
- **Before non-FIPS operation**—To begin repurposing your device for non-FIPS operation, perform zeroization before disabling FIPS mode of operation on the device or loading Junos OS packages that do not include FIPS mode of operation.

NOTE: **Juniper Networks does not support installing non-FIPS software in a FIPS mode of operation, but doing so might be necessary in certain test environments. Be sure to zeroize the system first.**

- **When a tamper-evident seal is disturbed**—If the seal on an insecure port has been tampered with, the system is considered to be compromised. After applying new tamper-evident seals to the appropriate locations, zeroize the system and set up new passwords and CSPs.

Related Documentation

- Applying Tamper-Evident Seals to the Cryptographic Module on page 37

Applying Tamper-Evident Seals to the Cryptographic Module

The cryptographic modules physical embodiment is that of a multi-chip standalone device that meets Level 2 physical security requirements. The module is completely enclosed in a rectangular nickel dor clear zinc coated, cold rolled steel, plated steel, and brushed aluminum enclosure. There are no ventilation holes, gaps, slits, cracks, slots, or crevices that would allow for any sort of observation of any component contained within the cryptographic boundary. Tamper-evident seals allow the operator to verify if the enclosure has been breached. These seals are not factory-installed and must be applied by the Cryptographic Officer.

NOTE: Seals are available for order from Juniper Networks using part number JNPR-FIPS-TAMPER-LBLS.

As a Cryptographic Officer, you are responsible for:

- Applying seals to secure the cryptographic module
- Controlling any unused seals
- Controlling and observing any changes, such as repairs or booting from an external USB drive to the cryptographic module, that require removing or replacing the seals to maintain the security of the module

As per the security inspection guidelines, upon receipt of the cryptographic module, the Cryptographic Officer must check that the labels are free of any tamper evidence.

- General Tamper-Evident Seal Instructions on page 37
- Applying Tamper-Evident Seals on the SRX4600 Device on page 37

General Tamper-Evident Seal Instructions

All FIPS-certified switches require a tamper-evident seal on the USB ports. While applying seals, follow these general instructions:

- Handle the seals with care. Do not touch the adhesive side. Do not cut or otherwise resize a seal to make it fit.
- Make sure all surfaces to which the seals are applied are clean and dry and clear of any residue.
- Apply the seals with firm pressure across the seal to ensure adhesion. Allow at least 1 hour for the adhesive to cure.

Applying Tamper-Evident Seals on the SRX4600 Device

Front and rear view:

- The front of the device has four HA ports, one USB port and two SSD slots that require tamper-evident label application. Four tamper-evident labels are placed over the HA

ports. Two tamper-evident labels are placed over the USB port so that the port and the screw are covered. The SSD will each have one tamper-evident label.

- The module contains two swappable PSU. Apply two tamper-evident labels on the two swappable PSUs.

Left and right view:

- On the left side of the device, apply one tamper-evident label over the fourth screw from the front.
- The right side of the device, apply one tamper-evident label over the fourth screw from the front.

Related Documentation

- How to Enable and Configure Junos OS in FIPS Mode of Operation on page 38

Loading Firmware on the Device

The Junos OS 18.1R1 FIPS images only accept the firmware signed with ECDSA and rejects any firmware signed with RSA+SHA1. You cannot downgrade to images that are signed with RSA+SHA1 from "ECDSA signed only" images. In this scenario, the SRX4600 device does not load the firmware.

Related Documentation

- How to Enable and Configure Junos OS in FIPS Mode of Operation on page 38

How to Enable and Configure Junos OS in FIPS Mode of Operation

You, as Cryptographic Officer, can enable and configure Junos OS in FIPS mode of operation on your device. Before you begin enabling and configuring FIPS mode of operation on the device:

- Verify the secure delivery of your device. See "Identifying Secure Product Delivery" on page 22.
- Apply tamper-evident seals. See "Applying Tamper-Evident Seals to the Cryptographic Module" on page 37.

To enable the Junos OS in FIPS mode of operation, perform the following steps:

1. Zeroize the device before enabling FIPS mode of operation

   user@host> request system zerioze hypervisor

2. Enable the FIPS mode on the device.

   user@host# set system fips level 2

3. Remove the CSPs on commit check and reboot the device.

   user@host# commit

4. Run integrity and self-tests on powering on the device when the module is operating in FIPS mode.

5. Configure IKEv2 when AES-GCM is used for encryption of IKE and/or IPSec.

```
user@host# set security ike proposal <ike_proposal_name> encryption-algorithm ?
Possible completions:
3des-cbc 3DES-CBC encryption algorithm
aes-128-cbc AES-CBC 128-bit encryption algorithm
aes-128-gcm AES-GCM 128-bit encryption algorithm
aes-192-cbc AES-CBC 192-bit encryption algorithm
aes-256-cbc AES-CBC 256-bit encryption algorithm
aes-256-gcm AES-GCM 256-bit encryption algorithm
user@host# set security ike proposal <ike_proposal_name> encryption-algorithm
  aes-256-gcm
user@host# set security ipsec proposal <ipsec_proposal_name> encryption-algorithm
  ?
Possible completions:
3des-cbc 3DES-CBC encryption algorithm
aes-128-cbc AES-CBC 128-bit encryption algorithm
aes-128-gcm AES-GCM 128-bit encryption algorithm
aes-192-cbc AES-CBC 192-bit encryption algorithm
aes-192-gcm AES-GCM 192-bit encryption algorithm
aes-256-cbc AES-CBC 256-bit encryption algorithm
aes-256-gcm AES-GCM 256-bit encryption algorithm
user@host# set security ipsec proposal <ipsec_proposal_name> encryption-algorithm
  aes-128-gcm
user@host# set security ike gateway <gateway_name> version ?
Possible completions:
v1-only The connection must be initiated using IKE version 1
v2-only The connection must be initiated using IKE version 2
user@host# set security ike gateway <gateway_name> version v2-only
user@host# commit
commit complete
```

6. Ensure that the backup image of the firmware is also a JUNOS-FIPS image by issuing the **request system snapshot** command.

NOTE: The show configuration security ike and show configuration security ipsec commands display the approved and configured IKE/IPsec configuration for the device operating in FIPS-approved mode.

```
root@host-srx4600> show version
Hostname: host-srx4600
Model: srx4600
Junos: 18.1R1.9
JUNOS OS Kernel 64-bit  [20180308.0604c57_builder_stable_11]
JUNOS OS libs [20180308.0604c57_builder_stable_11]
JUNOS OS runtime [20180308.0604c57_builder_stable_11]
JUNOS OS time zone information [20180308.0604c57_builder_stable_11]
JUNOS OS libs compat32 [20180308.0604c57_builder_stable_11]
JUNOS OS 32-bit compatibility [20180308.0604c57_builder_stable_11]
JUNOS py extensions [20180323.181821_builder_junos_181_r1]
JUNOS py base [20180323.181821_builder_junos_181_r1]
JUNOS OS vmguest [20180308.0604c57_builder_stable_11]
JUNOS OS crypto [20180308.0604c57_builder_stable_11]
JUNOS network stack and utilities [20180323.181821_builder_junos_181_r1]
JUNOS libs [20180323.181821_builder_junos_181_r1]
JUNOS libs compat32 [20180323.181821_builder_junos_181_r1]
JUNOS runtime [20180323.181821_builder_junos_181_r1]
JUNOS Web Management Platform Package
[20180323.181821_builder_junos_181_r1]
JUNOS srx libs compat32 [20180323.181821_builder_junos_181_r1]
JUNOS srx runtime [20180323.181821_builder_junos_181_r1]
JUNOS srx platform support [20180323.181821_builder_junos_181_r1]
JUNOS common platform support [20180323.181821_builder_junos_181_r1]
JUNOS srxtvp runtime [20180323.181821_builder_junos_181_r1]
JUNOS pppoe [20180323.181821_builder_junos_181_r1]
JUNOS mtx network modules [20180323.181821_builder_junos_181_r1]
JUNOS modules [20180323.181821_builder_junos_181_r1]
JUNOS srx modules [20180323.181821_builder_junos_181_r1]
JUNOS srxtvp libs [20180323.181821_builder_junos_181_r1]
JUNOS srx libs [20180323.181821_builder_junos_181_r1]
JUNOS srx Data Plane Crypto Support
[20180323.181821_builder_junos_181_r1]
JUNOS daemons [20180323.181821_builder_junos_181_r1]
JUNOS srx daemons [20180323.181821_builder_junos_181_r1]
JUNOS Extension Toolkit [20180323.181821_builder_junos_181_r1]
JUNOS Online Documentation [20180323.181821_builder_junos_181_r1]
JUNOS jail runtime [20180308.0604c57_builder_stable_11]
JUNOS fips optest i386 [20180323.181821_builder_junos_181_r1]
JUNOS FIPS mode utilities [20180323.181821_builder_junos_181_r1]
```

The fips keyword next to the hostname in the output indicates that the module is operating in FIPS mode for Junos Software Release 18.1R1.

```
user@host-srx4600:fips> show configuration security ike
proposal ike-proposal1 {
    authentication-method pre-shared-keys;
    dh-group group14;
    encryption-algorithm aes-256-gcm;
}
policy ike-policy1 {
    mode main;
```

```
    proposals ike-proposal1;
    pre-shared-key ascii-text "$9$Hq.5zF/tpBUj9AuOIRdbwsaZ"; ##
SECRET-DATA
}
gateway gw1 {
    ike-policy ike-policy1;
    address 198.51.100.0;
    local-identity inet 203.0.113.0;
    external-interface xe-0/0/3;
    version v2-only;
}
```

```
user@host-srx4600:fips> show configuration security ipsec
proposal ipsec-proposal1 {
    protocol esp;
    encryption-algorithm aes-128-gcm;
}
policy ipsec-policy1 {
    perfect-forward-secrecy {
        keys group14;
    }
    proposals ipsec-proposal1;
}
vpn vpn1 {
    bind-interface st0.0;
    ike {
        gateway gw1;
        ipsec-policy ipsec-policy1;
    }
}
```

Related Documentation

- Loading Firmware on the Device on page 38

CHAPTER 4

Configuring SSH and Console Connection

Understanding FIPS Authentication Methods

The Juniper Networks Junos operating system (Junos OS) running in FIPS mode of operation allows a wide range of capabilities for users, and authentication is identity-based. The following types of identity-based authentication are supported in the FIPS mode of operation:

Username and Password Authentication over the Console and SSH

In this authentication method, the user is requested to enter the username and password. The device enforces the user to enter a minimum of 10 characters password that is chosen from the 96 human-readable ASCII characters.

NOTE: **The maximum password length is 20 characters.**

In this method, the device enforces a timed access mechanism—for example, first two failed attempts to enter the correct password (assuming 0 time to process), no timed access is enforced. When the user enters the password for the third time, the module enforces a 5 second delay. Each failed attempt thereafter results in an additional 5 second delay above the previous failed attempt. For example, if the fourth failed attempt is a 10 second delay, then the fifth failed attempt is a 15 second delay, the sixth failed attempt is a 20 second delay, and the seventh failed attempt is a 25 second delay.

Therefore, this leads to a maximum of seven possible attempts in a 1 minute period for each getty active terminal. So, the best approach for the attacker would be to disconnect after 4 failed attempts, and wait for a new getty to be spawned. This would allow the attacker to perform roughly 9.6 attempts per minute (576 attempts per hour or 60

minutes). This would be rounded off to 9 attempts per minute, because there is no such thing as 0.6 attempts. Thus the probability of a successful random attempt is 1/9610, which is less than 1/1 million. The probability of a success with multiple consecutive attempts in a 1 minute period is 9/(9610), which is less than 1/100,000.

Username and Public Key Authentication over SSH

In SSH public key authentication, you provide the username and validate the ownership of the private key corresponding to the public key stored on the server. The device supports ECDSA (P-256, P-384, and P-521) and RSA (2048, 3072, and 4092 modulus bit length) key-types. The probability of a success with multiple consecutive attempts in a 1-minute period is 5.6e7/(2128).

Related Documentation

- Configuring SSH on the Evaluated Configuration on page 46

Configuring a System Login Message and Announcement

A system login message appears before the user logs in and a system login announcement appears after the user logs in. By default, no login message or announcement is displayed on the device.

To configure a system login message, use the following command:

```
[edit]
user@host# set system login message login-message-banner-text
```

To configure system announcement, use the following command:

```
[edit]
user@host# set system login announcement system-announcement-text
```

NOTE:

- If the message text contains any spaces, enclose it in quotation marks.
- You can format the message using the following special characters:
  - \n—New line
  - \t—Horizontal tab
  - \'—Single quotation mark
  - \"—Double quotation mark
  - \\—Backslash

Related Documentation

- Configuring SSH on the Evaluated Configuration on page 46

Limiting the Number of User Login Attempts for SSH Sessions

A remote administrator may login to a device through SSH. Administrator credentials are stored locally on the device. If the remote administrator presents a valid username and password, access to the TOE is granted. If the credentials are invalid, the TOE allows the authentication to be retried after an interval that starts after 1 second and increases exponentially. If the number of authentication attempts exceed the configured maximum, no authentication attempts are accepted for a configured time interval. When the interval expires, authentication attempts are again accepted.

You can configure the device to limit the number of attempts to enter a password while logging through SSH. Using the following command, the connection can be terminated if a user fails to login after a specified number of attempts:

```
[edit system login]
user@host# set retry options tries-before-disconnect <number>
```

Here, **tries-before-disconnect** is the number of times a user can attempt to enter a password when logging in. The connection closes if a user fails to log in after the number specified. The range is from 1 through 10, and the default value is 10.

You can also configure a delay, in seconds, before a user can try to enter a password after a failed attempt.

```
[edit system login]
user@host# set retry options backoff-threshold <number>
```

Here, **backoff-threshold** is the threshold for the number of failed login attempts before the user experiences a delay in being able to enter a password again. Use the **backoff-factor** option to specify the length of the delay in seconds. The range is from 1 through 3, and the default value is 2 seconds.

In addition, the device can be configured to specify the threshold for the number of failed attempts before the user experiences a delay in entering the password again.

```
[edit system login]
user@host# set retry options backoff-factor <number>
```

Here, **backoff-factor** is the length of time, in seconds, before a user can attempt to log in after a failed attempt. The delay increases by the value specified for each subsequent attempt after the threshold. The range is from 5 through 10, and the default value is 5 seconds.

The SSHv2 protocol provides secure terminal sessions utilizing the secure encryption. The SSHv2 protocol enforces running the key-exchange phase and changing the encryption and integrity keys for the session. Key exchange is done periodically, after specified seconds or after specified bytes of data have passed over the connection. You can configure thresholds for SSH rekeying, (FCS_SSHS_EXT.1.8 and FCS_SSHC_EXT.1.8). The TSF ensures that within the SSH connections the same session keys are used for a

threshold of no longer than one hour, and no more than one gigabyte of the transmitted data. When either of the thresholds are reached, a rekey must be performed.

Related Documentation

- Configuring SSH on the Evaluated Configuration on page 46

Configuring SSH on the Evaluated Configuration

SSH is an allowed remote management interface in the evaluated configuration. This topic describes how to configure SSH on the device.

Before you begin, log in with your root account on the device running Junos OS Release 18.1R1 and edit the configuration.

NOTE: The commands shown configure SSH to use all of the allowed cryptographic algorithms.

NOTE: You can enter the configuration commands in any order and commit all the commands at once.

To configure SSH on the TOE:

1. Specify the permissible SSH host-key algorithms.

```
[edit system services ssh]
user@host# set hostkey-algorithm ssh-ecdsa
user@host# set hostkey-algorithm ssh-rsa
```

2. Specify the SSH key-exchange algorithms.

```
[edit system services ssh]
user@host#set key-exchange [ ecdh-sha2-nistp256 ecdh-sha2-nistp384
  ecdh-sha2-nistp521 ]
```

3. Specify all the permissible message authentication code algorithms.

```
[edit system services ssh]
user@host#set macs [ hmac-sha1 hmac-sha2-256 hmac-sha2-512 ]
```

4. Specify the ciphers allowed for protocol version 2.

```
[edit system services ssh]
user@host#set ciphers [ 3des-cbc aes128-cbc aes128-ctr aes192-cbc aes192-ctr
  aes256-cbc aes256-ctr ]
```

Related Documentation

- Limiting the Number of User Login Attempts for SSH Sessions on page 45

CHAPTER 5

Configuring the Remote Syslog Server

Sample Syslog Server Configuration on a Linux System

A secure Junos OS environment requires auditing of events and storing them in a local audit file. The recorded events are simultaneously sent to an external syslog server. A syslog server receives the syslog messages streamed from the device. The syslog server must have an SSH client with NETCONF support configured to receive the streamed syslog messages.

The NDcPP logs capture the events, few of them are listed below:

- Committed changes
- Login and logout of users
- Failure to establish an SSH session
- Establishment or termination of an SSH session
- Changes to the system time

The following procedure is an example to show how to configure a syslog server on a Linux platform using the StrongSwan configuration to provide IPsec. Before you begin, the Linux-based syslog server must be configured with the IP address and gateway, and the StrongSwan IPsec client must be installed on the syslog server to initiate a VPN connection with the Junos OS device.

To setup a StrongSwan configuration on the remote syslog server to provide IPsec VPN capability:

1. Modify the */etc/ipsec.secrets* settings in accordance with the Junos OS device configuration.

   ```
   root@host# vi /etc/ipsec.secrets 192.168.1.2 192.168.1.1 : PSK “12345”
   ```

2. Modify the */etc/ipsec.conf* settings in accordance with the Junos OS device configuration.

```
root@host# vi /etc/ipsec.conf
  config setup
                  charondebug="ike 4, cfg 4, chd 4, enc 1, net 4, knl 4, dmn
   4"
  conn %default
                ikelifetime=240
      keylife=300
      rekeymargin=10s
      keyingtries=%forever
      mobike=no
  conn home
                keyexchange=ikev1
      authby=psk
      ike=aes128-sha256-modp2048!
      esp=aes128-sha1-modp2048!
      left=192.168.1.2 # self if
      leftsubnet=203.0.113.1/24 # self net for proxy id
      leftid=192.168.1.2 # self id
      right=192.168.1.1 # peer if
      rightsubnet=192.168.2.0/24 # peer net for proxy id
      rightid=192.168.1.1 # peer id
      auto=add
      leftfirewall=yes
      dpdaction=restart
      dpddelay=10
      dpdtimeout=120
      rekeyfuzz=10%
      reauth=no
```

NOTE: Here **conn home** specifies the name of the IPSec tunnel connection to be established between a Junos OS device and Strongswan VPN Client on Syslog server, **ike=aes-sha256-modp2048** specifies the IKE encryption and authentication algorithms and DH Group to be used for the connection, and**esp=aes128-sha1** specifies the ESP encryption and authentication algorithms to be used for the connection.

3. Activate IPsec service by using **ipsec up <being-established-ipsec-tunnel-name>** command. For example,

```
[root@fipscc-pc02 regress]# ipsec up home
002 "home" #3: initiating Main Mode
104 "home" #3: STATE_MAIN_I1: initiate
010 "home" #3: STATE_MAIN_I1: retransmission; will wait 20s for response
```

4. Restart the IPsec StrongSwan service.

```
root@host# ipsec restart
```

5. Check for syslog encrypted traffic.

```
root@host# tcpdump –I eth1 –vv –s 1500 –c 10 –o /var/tmp/Syslog_Traffic.pcap
```

6. Copy **/var/log/syslog to /var/tmp/syslog_verify** file on the syslog server to validate the syslog from the Junos OS device.

```
root@host# cp /var/log/syslog /var/tmp/syslog_verify
```

Configuring Event Logging to a Local File

You can configure storing of audit information to a local file and the level of detail to be recorded with the **syslog** statement. This example stores logs in a file named **Audit_file**:

```
[edit system]
syslog {
   file Audit_file;
}
```

Configuring Event Logging to a Remote Server

Configure the export of audit information to a secure, remote server by setting up an event trace monitor that sends event log messages by using NETCONF over SSH to the remote system event logging server. The following procedures show the configuration needed to send system log messages to a secure external server by using NETCONF over SSH.

Configuring Event Logging to a Remote Server when Initiating the Connection from the Remote Server

The following procedure describes the steps to configure event logging to a remote server when the SSH connection to the TOE is initiated from the remote system log server.

1. Generate an RSA public key on the remote syslog server.

```
$ ssh-keygen -b 2048 -t rsa -C 'syslog-monitor key pair' -f ~/.ssh/syslog-monitor
```

   You will be prompted to enter the desired passphrase. The storage location for the **syslog-monitor** key pair is displayed.

2. On the TOE, create a class named **monitor** that has permission to trace events.

```
[edit]
user@host# set system login class monitor permissions trace
```

3. Create a user named **syslog-mon** with the class monitor, and with authentication that uses the **syslog-monitor** key pair from the key pair file located on the remote syslog server.

```
[edit]
user@host# set system login user syslog-mon class monitor authentication ssh-rsa
   public key from syslog-monitor key pair
```

4. Set up NETCONF with SSH.

```
[edit]
user@host# set system services netconf ssh
```

5. Configure syslog to log all the messages at */var/log/Audit_file*.

```
[edit]
user@host# set system syslog file Audit_file any any
user@host# commit
```

6. On the remote system log server, start up the SSH agent. The start up is required to simplify the handling of the syslog-monitor key.

```
$ eval `ssh-agent`
```

7. On the remote syslog server, add the **syslog-monitor** key pair to the SSH agent.

```
$ ssh-add ~/.ssh/syslog-monitor
```

You will be prompted to enter the desired passphrase. Enter the same passphrase used in Step 1.

8. After logging in to the **external_syslog_server** session, establish a tunnel to the device and start NETCONF.

```
user@host# ssh syslog-mon@NDcPP_TOE -s netconf > test.out
```

9. After NETCONF is established, configure a system log events message stream. This RPC will cause the NETCONF service to start transmitting messages over the SSH connection that is established.

```
<rpc><get-syslog-events><stream>messages</stream></get-syslog-events></rpc>
```

10. The examples for syslog messages are listed below. Monitor the event log generated for admin actions on TOE as received on the syslog server. Examine the traffic that passes between the audit server and the TOE, observing that these data are not viewed during this transfer, and that they are successfully received by the audit server. Match the logs between local event and the remote event logged in a syslog server and record the particular software (such as name, version, and so on) used on the audit server during testing.

The following output shows test log results for syslog server.

```
host@ssh-keygen -b 2048 -t rsa -C 'syslog-monitor key pair' -f
~/.ssh/syslog-monitor
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
Your identification has been saved in /home/host/.ssh/syslog-monitor.
Your public key has been saved in /home/host/.ssh/syslog-monitor.pub.
The key fingerprint is:
ef:75:d7:68:c5:ad:8d:6f:5e:7a:7e:9b:3d:f1:4d:3f syslog-monitor key pair
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|               ..|
|        S       +|
|         .     Bo|
|          . . *.X|
|         . . o E@|
|          .   .BX|
+-----------------+
[host@linux]$ cat /home/host/.ssh/syslog-monitor.pub
ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCrUREJUBpjwAoIgRrGy9zgt+
D2pikk3Q/Wdf8I5vr+njeqJhCx2bUAkrRbYXNILQQAZbg7kLfi/8TqqL
eon4HOP2e6oCSorKdx/GrOTzLONL4fhOEyuSAk8bs5JuwWNBUokVO25
gzpGFsBusGnlj6wqqJ/sjFsMmfxyCkbY+pUWb8m1/A9YjOFT+6esw+9S
tF6Gbg+VpbYYk/Oday4z+z7tQHRFSrxj2G92aoliVDBLJparEMBc8w
LdSUDxmgBTM2oadOmm+kreBUQjrmr6775RJn9H9YwIxKOxGm4SFnX/Vl4
R+lZ9RqmKH2wodIEM34KOwXEHzAzNZO1oLmaAVqT
syslog-monitor key pair
[host@linux]$ eval `ssh-agent `
Agent pid 1453
[host@linux]$ ssh-add ~/.ssh/syslog-monitor
Enter passphrase for /home/host/.ssh/syslog-monitor:
Identity added: /home/host/.ssh/syslog-monitor (/home/host/.ssh/syslog-monitor)
```

Net configuration channel

```
host@linux]$ ssh syslog-mon@starfire -s netconf>test.out
host@linux]$ cat test.out
this is NDcPP test device

<!-- No zombies were killed during the creation of this user interface --
<!-- user syslog-mon, class j-monitor -><hello>
  <capabilities>
    <capability>urn:ietf:params:xml:ns:netconf:base:1.0</capability>

<capability>urn:ietf:params:xml:ns:netconf:capability:candidate:1.0</capability>


<capability>urn:ietf:params:xml:ns:netconf:capability:confirmed-commit:1.0</capability>

   <capability>urn:ietf:params:xml:ns:netconf:capability:validate:1.0</capability>


<capability>urn:ietf:params:xml:ns:netconf:capability:url:1.0?protocol=http,ftp,file</capability>

    <capability>http://xml.juniper.net/netconf/junos/1.0</capability>
    <capability>http://xml.juniper.net/dmi/system/1.0</capability>
  </capabilities>
  <session-id4129/session-id>
```

```
</hello>
]]>]]>
```

The following output shows event logs generated on the TOE that are received on the syslog server.

```
Jan 20 17:04:51  starfire sshd[4182]: error: Could not load host key:
/etc/ssh/ssh_host_dsa_key
Jan 20 17:04:51  starfire sshd[4182]: error: Could not load host key:
/etc/ssh/ssh_host_ecdsa_key
Jan 20 17:04:53  starfire sshd[4182]: Accepted password for sec-admin from
10.209.11.24 port 55571 ssh2
Jan 20 17:04:53  starfire mgd[4186]: UI_AUTH_EVENT: Authenticated user 'sec-admin'
 at permission level 'j-administrator'
Jan 20 17:04:53  starfire mgd[4186]: UI_LOGIN_EVENT: User 'sec-admin' login, class
 'j-administrator' [4186], ssh-connection '10.209.11.24 55571 10.209.14.92 22',
client-mode 'cli'
```

Net configuration channel

```
host@linux]$ ssh syslog-mon@starfire -s netconf
 this is NDcPP test device

<!-- No zombies were killed during the creation of this user interface --
<!-- user syslog-mon, class j-monitor -><hello>
  <capabilities>
    <capability>urn:ietf:params:xml:ns:netconf:base:1.0</capability>

<capability>urn:ietf:params:xml:ns:netconf:capability:candidate:1.0</capability>


<capability>urn:ietf:params:xml:ns:netconf:capability:confirmed-commit:1.0</capability>

   <capability>urn:ietf:params:xml:ns:netconf:capability:validate:1.0</capability>


<capability>urn:ietf:params:xml:ns:netconf:capability:url:1.0?protocol=http,ftp,file</capability>

    <capability>http://xml.juniper.net/netconf/junos/1.0</capability>
    <capability>http://xml.juniper.net/dmi/system/1.0</capability>
  </capabilities>
  <session-id4129/session-id>
</hello>
]]>]]>
```

The following output shows that the local syslogs and remote syslogs received are similar.

```
Local : an 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation
 in progress: Redundancy interface management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/rdd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/rdd', PID 4317, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
```

```
progress: Dynamic flow capture service checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child
'/usr/sbin/dfcd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/dfcd', PID 4318, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Connectivity fault management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child
'/usr/sbin/cfmd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/cfmd', PID 4319, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Layer 2 address flooding and learning process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child
'/usr/sbin/l2ald'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/l2ald', PID 4320, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Layer 2 Control Protocol process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child
'/usr/sbin/l2cpd'
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines complete
Jan 20 17:09:30  starfire l2cp[4321]: Initialized 802.1X module and state
machinesJan 20 17:09:30  starfire l2cp[4321]: Read acess profile () config
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/l2cpd', PID 4321, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Multicast Snooping process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child
'/usr/sbin/mcsnoopd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/mcsnoopd', PID 4325, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: commit wrapup...
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: activating '/var/etc/ntp.conf'
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: start ffp activate
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/ffp'
Jan 20 17:09:30  starfire ffp[4326]: "dynamic-profiles": No change to
profiles.....................................
```

```
Remote : an 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation
 in progress: Redundancy interface management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/rdd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/rdd', PID 4317, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Dynamic flow capture service checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child
'/usr/sbin/dfcd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/dfcd', PID 4318, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Connectivity fault management process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child
'/usr/sbin/cfmd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
```

```
'/usr/sbin/cfmd', PID 4319, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Layer 2 address flooding and learning process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child
'/usr/sbin/l2ald'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/l2ald', PID 4320, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Layer 2 Control Protocol process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child
'/usr/sbin/l2cpd'
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines
Jan 20 17:09:30  starfire l2cp[4321]: Initializing PNAC state machines complete
Jan 20 17:09:30  starfire l2cp[4321]: Initialized 802.1X module and state
machinesJan 20 17:09:30  starfire l2cp[4321]: Read acess profile () config
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/l2cpd', PID 4321, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: Multicast Snooping process checking new configuration
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child
'/usr/sbin/mcsnoopd'
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_STATUS: Cleanup child
'/usr/sbin/mcsnoopd', PID 4325, status 0
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: commit wrapup...
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: activating '/var/etc/ntp.conf'
Jan 20 17:09:30  starfire mgd[4186]: UI_COMMIT_PROGRESS: Commit operation in
progress: start ffp activate
Jan 20 17:09:30  starfire mgd[4186]: UI_CHILD_START: Starting child '/usr/sbin/ffp'
Jan 20 17:09:30  starfire ffp[4326]: "dynamic-profiles": No change to profiles
..............
```

Forwarding Logs to the External Syslog Server

When the device running Junos OS is set up for an external syslog server, the TOE forwards copies of local logs to the external syslog server and retains local copies of all logs when the TOE is configured in event log mode. In stream log mode, all logs except traffic logs are stored locally and can be forwarded to an external syslog server, whereas traffic logs can only be forwarded to an external syslog server.

The connection between the device running Junos OS and the syslog server is established on an event basis depending on preconfiguration of what type of logs are forwarded from local to external. When the configured condition is met, the device sends local logs to the external syslog server.

Related Documentation

- Sample Syslog Server Configuration on a Linux System on page 49

CHAPTER 6

Configuring Audit Log Options

Configuring Audit Log Options in the Evaluated Configuration

The following section describes how to configure audit log options in the evaluated configuration.

Configuring Audit Log Options for SRX4600 Devices

To configure audit log options for SRX4600 devices:

1. Specify the number of files to be archived in the system logging facility.

```
[edit system syslog]
root@host#set archive files 2
```

2. Specify the file in which to log data.

```
[edit system syslog]
root@host#set file syslog any any
```

3. Specify the size of files to be archived.

```
[edit system syslog]
root@host#set file syslog archive size 10000000
```

4. Specify the priority and facility in messages for the system logging facility.

```
[edit system syslog]
root@host#set file syslog explicit-priority
```

5. Log system messages in a structured format.

```
[edit system syslog]
root@host#set file syslog structured-data
```

6. Specify how security logs need to be processed and exported.

```
[edit]
root@host#set security log mode event
```

Related Documentation

- Sample Code Audits of Configuration Changes on page 58

Sample Code Audits of Configuration Changes

This sample code audits all changes to the configuration secret data and sends the logs to a file named **Audit-File:**

```
[edit system]
syslog {
   file Audit-File {
      authorization info;
      change-log info;
      interactive-commands info;
   }
}
```

This sample code expands the scope of the minimum audit to audit all changes to the configuration, not just secret data, and sends the logs to a file named **Audit-File:**

```
[edit system]
syslog {
   file Audit-File {
      any any;
      authorization info;
      change-log any;
      interactive-commands info;
      kernel info;
      pfe info;
   }
}
```

Example: System Logging of Configuration Changes

This example shows a sample configuration and makes changes to users and secret data. It then shows the information sent to the audit server when the secret data is added to the original configuration and committed with the **load** command.

```
[edit system]
location {
   country-code US;
   building B1;
}
...
```

```
login {
   message "UNAUTHORIZED USE OF THIS ROUTER\n\tIS STRICTLY PROHIBITED!";
      user admin {
         uid 2000;
          class super-user;
      authentication {
         encrypted-password “$ABC123”;
         # SECRET-DATA
      }
   }
   password {
      format md5;
   }
}
radius-server 192.0.2.15 {
   secret “$ABC123” # SECRET-DATA
}
services {
   ssh;
}
syslog {
   user *{
      any emergency;
   }
   file messages {
      any notice;
      authorization info;
   }
   file interactive-commands {
      interactive-commands any;
   }
}
...
...
```

The new configuration changes the secret data configuration statements and adds a new user.

```
user@host# show | compare
[edit system login user admin authentication]
-    encrypted-password “$ABC123”; # SECRET-DATA
+    encrypted-password “$ABC123”; # SECRET-DATA
[edit system login]
+    user admin2 {
+        uid 2001;
+        class operator;
+        authentication {
+            encrypted-password “$ABC123”;
                # SECRET-DATA
+        }
+    }
[edit system radius-server 192.0.2.15]
-    secret “$ABC123”; # SECRET-DATA
+    secret “$ABC123”; # SECRET-DATA
```

Related Documentation

- Configuring Audit Log Options in the Evaluated Configuration on page 57

CHAPTER 7

Configuring Event Logging

Event Logging Overview

The evaluated configuration requires the auditing of configuration changes through the system log.

In addition, Junos OS can:

- Send automated responses to audit events (syslog entry creation).
- Allow authorized managers to examine audit logs.
- Send audit files to external servers.
- Allow authorized managers to return the system to a known state.

The logging for the evaluated configuration must capture the following events:

- Changes to secret key data in the configuration.
- Committed changes.
- Login/logout of users.
- System startup.
- Failure to establish an SSH session.
- Establishment/termination of an SSH session.
- Changes to the (system) time.
- Termination of a remote session by the session locking mechanism.
- Termination of an interactive session.

In addition, Juniper Networks recommends that logging also:

- Capture all changes to the configuration.
- Store logging information remotely.

Related Documentation

- Interpreting Event Messages on page 62

Interpreting Event Messages

The following output shows a sample event message.

```
Jul 24 17:43:28 router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system
    radius-server 1.2.3.4 secret]
```

Table 6 on page 62 describes the fields for an event message. If the system logging utility cannot determine the value in a particular field, a hyphen (-) appears instead.

Table 6: Fields in Event Messages

Field	Description	Examples
timestamp	Time when the message was generated, in one of two representations: • ***MMM-DD HH:MM:SS.MS+/-HH:MM,*** is the month, day, hour, minute, second and millisecond in local time. The hour and minute that follows the plus sign (+) or minus sign (-) is the offset of the local time zone from Coordinated Universal Time (UTC). • ***YYYY-MM-DDTHH:MM:SS.MSZ*** is the year, month, day, hour, minute, second and millisecond in UTC.	Jul 24 17:43:28 is the timestamp expressed as local time in the United States. 2012-07-24T09:17:15.719Z is 9:17 AM UTC on 24 July 2012.
hostname	Name of the host that originally generated the message.	router1
process	Name of the Junos OS process that generated the message.	mgd
processID	UNIX process ID (PID) of the Junos OS process that generated the message.	4153
TAG	Junos OS system log message tag, which uniquely identifies the message.	UI_DBASE_LOGOUT_EVENT
username	Username of the user initiating the event.	“admin”
message-text	English-language description of the event .	set: [system radius-server 1.2.3.4 secret]

Related Documentation

- Event Logging Overview on page 61

Logging Changes to Secret Data

The following are examples of audit logs of events that change the secret data.

Load Merge

When a **load merge** command is issued to merge the contents of the example Common Criteria configuration with the contents of the original configuration, the following audit logs are created concerning the secret data:

```
Jul 24 17:43:28  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system radius-server
1.2.3.4 secret]
Jul 24 17:43:28  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login user admin
authentication encrypted-password]
Jul 24 17:43:28  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login user admin2
 authentication encrypted-password]
```

Load Replace

When a **load replace** command is issued to replace the contents of the example Common Criteria configuration with the contents of the original configuration, the following audit logs are created concerning the secret data:

```
Jul 24 18:29:09  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' replace: [system radius-server
 1.2.3.4 secret]
Jul 24 18:29:09  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' replace: [system login user
admin authentication encrypted-password]
Jul 24 18:29:09  router1 mgd[4163]: UI_CFG_AUDIT_SET_SECRET: User 'admin' replace: [system login user
admin authentication encrypted-password]
```

Load Override

When a **load override** command is issued to override the contents of the example Common Criteria configuration with the contents of the original configuration, the following audit logs are created concerning the secret data:

```
Jul 25 14:25:51  router1 mgd[4153]: UI_LOAD_EVENT: User 'admin' is performing a 'load override'
Jul 25 14:25:51  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' override: CC_config2.txt
Jul 25 14:25:51  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system radius-server
1.2.3.4 secret]
Jul 25 14:25:51  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login user admin
authentication encrypted-password]
Jul 25 14:25:51  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login user admin
authentication encrypted-password]
```

Load Update

When a **load update** command is issued to update the contents of the example Common Criteria configuration with the contents of the original configuration, the following audit logs are created concerning the secret data:

```
Jul 25 14:31:03  router1 mgd[4153]: UI_LOAD_EVENT: User 'admin' is performing a 'load update'
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' update: CC_config2.txt
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system radius-server
1.2.3.4 secret]
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' deactivate: [system radius-server
1.2.3.4 secret] ""
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login user admin
authentication encrypted-password]
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' deactivate: [system login user admin
 authentication encrypted-password] ""
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_SET_SECRET: User 'admin' set: [system login user test
authentication encrypted-password]
Jul 25 14:31:03  router1 mgd[4153]: UI_CFG_AUDIT_OTHER: User 'admin' deactivate: [system login user test
 authentication encrypted-password] ""
```

For more information about configuring parameters and managing log files, see the *Junos OS System Log Messages Reference*.

Related Documentation

- Forwarding Logs to the External Syslog Server on page 56
- Interpreting Event Messages on page 62

Login and Logout Events Using SSH

System log messages are generated whenever a user successfully or unsuccessfully attempts SSH access. Logout events are also recorded. For example, the following logs are the result of two failed authentication attempts, then a successful one, and finally a logout:

```
Dec 20 23:17:35  bilbo sshd[16645]: Failed password for op from 172.17.58.45 port 1673 ssh2
Dec 20 23:17:42  bilbo sshd[16645]: Failed password for op from 172.17.58.45 port 1673 ssh2
Dec 20 23:17:53  bilbo sshd[16645]: Accepted password for op from 172.17.58.45 port 1673 ssh2
Dec 20 23:17:53  bilbo mgd[16648]: UI_AUTH_EVENT: Authenticated user 'op' at permission level
                                   'j-operator'
Dec 20 23:17:53  bilbo mgd[16648]: UI_LOGIN_EVENT: User 'op' login, class 'j-operator' [16648]
Dec 20 23:17:56  bilbo mgd[16648]: UI_CMDLINE_READ_LINE: User 'op', command 'quit '
Dec 20 23:17:56  bilbo mgd[16648]: UI_LOGOUT_EVENT: User 'op' logout
```

Related Documentation

- Interpreting Event Messages on page 62

Logging of Audit Startup

The audit information logged includes startups of Junos OS. This in turn identifies the startup events of the audit system, which cannot be independently disabled or enabled. For example, if Junos OS is restarted, the audit log contains the following information:

```
Dec 20 23:17:35  bilbo syslogd: exiting on signal 14
Dec 20 23:17:35  bilbo syslogd: restart
Dec 20 23:17:35  bilbo syslogd /kernel: Dec 20 23:17:35 init: syslogd (PID 19128) exited with status=1
Dec 20 23:17:42  bilbo /kernel:
Dec 20 23:17:53  init: syslogd (PID 19200) started
```

Related Documentation

- Login and Logout Events Using SSH on page 64

CHAPTER 8

Configuring a Secure Logging Channel

Creating a Secure Logging Channel

This section describes how to place the device in an evaluated configuration to provide an encrypted communication channel over an IPsec VPN tunnel, between a device running Junos OS and a remote external storage server (syslog server).

NOTE: **The ssh-rsa authentication method is one of the allowed algorithms in FIPS mode.**

Table 7 on page 67 lists all the supported algorithms for the IPsec VPN tunnel.

Table 7: IPsec VPN Tunnel Supported Algorithms

IKE Phase1 Proposal			
Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
pre-shared-keys	sha-256	group14	aes-128-cbc
rsa-signatures-2048	sha-384	group19	aes-128-gcm
ecdsa-signatures-256		group20	aes-192-cbc
ecdsa-signatures-384		group24	aes-256-cbc
			aes-256-gcm
			3des-cbc

IPSec Phase2 Proposal			
Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
hmac-sha1-96	group14	ESP	aes-128-cbc
hmac-sha-256-128	group19		aes-128-gcm
	group20		aes-192-cbc
	group24		aes-192-gcm
			aes-256-cbc
			aes-256-gcm
			3des-cbc

d

Configuring a Trusted Path or Channel Between a Device Running Junos OS and a Remote External Storage Server

This section describes the configuration details required to provide an encrypted communication channel between a device running Junos OS and the remote external storage server through an IPsec VPN tunnel.

NOTE: The remote external storage server is a Linux-based syslog server on which the IPsec VPN Tunnel is terminated at the outbound interface Eth1. The log data transferred from the device is sent to the syslog termination interface Eth2 and the StrongSwan application to provide the IPsec VPN capability.

Table 8 on page 68 lists the IPsec VPN tunnel details used in this example.

Table 8: IPsec VPN Tunnel Information

Phase 1 Proposal (P1, IKE)				Phase 2 Proposal (P2, IPSec)			
Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm	Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
pre-shared-keys	sha-256	group14	aes-128-cbc	hmac-sha1-96	group14	ESP	aes-128-cbc

Figure 1 on page 69 illustrates the encrypted communication channel between a device running Junos OS and a remote external storage server. An IPsec tunnel is established between a devices egress interface (Intf-1) and a remote syslog server outbound interface (Eth1). Data is then forwarded internally on the remote external storage server from its outbound interface Eth1; that is, the VPN endpoint to Eth2.

Figure 1: IPsec VPN Tunnel

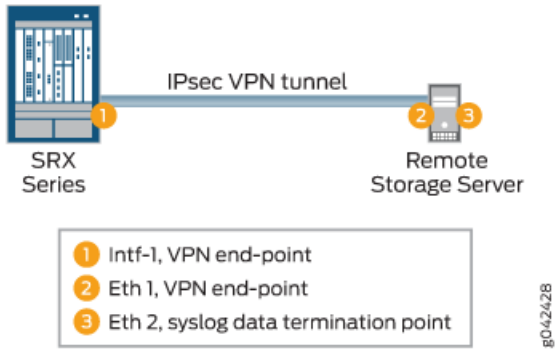


Table 9 on page 69 provides the interface and IP configuration details used in this example.

Table 9: Interface and IP Configuration Details for the Trusted Path

Device Running Junos OS	Remote Storage Server
IP Address: "Intf-2" interface: XE-0/0/1 – IP Address: 198.51.100.2 "Intf-1" interface: XE-0/0/2 - IP Address: 198.51.100.1 Enable: Syslog logging to remote syslog server	IP Address: Eth1: 198.51.100.3 Eth2: 203.0.113.1 Gateway Eth1: 198.51.100.1 Tools: SSH and Strongswan (for IPsec VPN)

To configure the trusted path or channel between a device running Junos OS and a remote external storage server:

1. Enable stream logging for traffic logs.

```
[edit security]
user@host#set log cache
user@host#set log mode event
user@host#set log source-address 198.51.100.2
user@host#set log stream STREAM category all
user@host#set log stream STREAM host 203.0.113.1
```

NOTE: 192.168.2.1 is the IP address of the syslog server outbound interface at which the IPsec VPN tunnel is terminated, and 20.20.20.2 is the IP address of the syslog server interface for which log data is destined.

2. Enable syslog on the device.

```
[edit system]
```

```
user@host#set syslog user * any emergency
user@host#set syslog host 203.0.113.1 any any
user@host#set syslog file SYSLOG any any
user@host#set syslog file SYSLOG authorization info
user@host#set syslog file SYSLOG_COMMANDS interactive-commands error
user@host#set syslog file traffic-log any any
user@host#set syslog file traffic-log match RT_FLOW_SESSION
user@host#set syslog source-address 198.51.100.2
```

3. Enable VPN on the device.

IKE setup:

```
[edit security]
user@host#set ike proposal IKE_Proposal authentication-method pre-shared-keys
user@host#set ike proposal IKE_Proposal dh-group group14
user@host#set ike proposal IKE_Proposal authentication-algorithm sha-256
user@host#set ike proposal IKE_Proposal encryption-algorithm aes-128-cbc
user@host#set ike policy IKE_Policy mode main
user@host#set ike policy IKE_Policy proposals IKE_Proposal
user@host#prompt ike policy IKE_Policy pre-shared-key ascii-text 12345
user@host#set ike gateway GW ike-policy IKE_Policy
user@host#set ike gateway GW address 198.51.100.3
user@host#set ike gateway GW local-identity inet 198.51.100.1
user@host#set ike gateway GW external-interface xe-0/0/2
user@host#set ike gateway GW version v2-only
```

IPsec setup:

```
[edit security ipsec]
user@host#set proposal IPsec_Proposal protocol esp
root@host#set proposal IPsec_Proposal authentication-algorithm hmac-sha1-96
root@host#set proposal IPsec_Proposal encryption-algorithm aes-128-cbc
root@host#set policy IPsec_Policy perfect-forward-secrecy keys group14
root@host#set policy IPsec_Policy proposals IPsec_Proposal
root@host#set vpn VPN bind-interface st0.0
root@host#set vpn VPN ike gateway GW
root@host#set vpn VPN ike ipsec-policy IPsec_Policy
root@host#set vpn VPN establish-tunnels immediately
```

4. Perform the following additional configurations on the device.

IKE trace log:

```
[edit security ike
root@host#set traceoptions file IKE_Trace
root@host#set traceoptions file size 10000000
root@host#set ike traceoptions flag all
```

Flow trace:

```
[edit security flow ]
```

```
root@host#set traceoptions file DEBUG
root@host#set traceoptions file size 1000000
root@host#set traceoptions flag all
```

Route options:

```
[edit ]
root@host#set routing-options static route 203.0.113.2/24 qualified-next-hop st0.0
  preference 1
```

Address book configuration:

```
[edit security address-book]
root@host#set global address trustLAN 198.51.100.0/24
root@host#set global address unTrustLAN 198.51.100.3/24
```

Zone configuration:

```
[edit security zones]
root@host#set trustZone host-inbound-traffic system-services all
root@host#set security-zone trustZone host-inbound-traffic protocols all
root@host#set security-zone trustZone interfaces xe-0/0/1.0
root@host#set security-zone unTrustZone host-inbound-traffic system-services all
root@host#set security-zone unTrustZone host-inbound-traffic protocols all
root@host#set security-zone unTrustZone interfaces st0.0
root@host#set security-zone unTrustZone interfaces xe-0/0/2.0
```

Policy configuration:

```
[edit security policies]
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 match
  source-address trustLAN
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 match
  destination-address unTrustLAN
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 match
  application any
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 then permit
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 then log
  session-init
root@host#set from-zone trustZone to-zone unTrustZone policy Policy1 then log
  session-close
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 match
  source-address unTrustLAN
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 match
  destination-address trustLAN
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 match
  application any
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 then permit
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 then log
  session-init
root@host#set from-zone unTrustZone to-zone trustZone policy Policy1 then log
  session-close
```

Related Documentation

- Configuring SSH on the Evaluated Configuration on page 46
- *Sample Syslog Server Configuration on a Linux System*

CHAPTER 9

Configuring VPNs

Configuring VPN on a Device Running Junos OS

This section describes sample configurations of an IPsec VPN on a Junos OS device using the following IKE authentication methods:

Figure 2 on page 73 illustrates the VPN topology used in all the examples described in this section. Here, H0 and H1 are the host PCs, R0 and R2 are the two endpoints of the IPsec VPN tunnel, and R1 is a router to route traffic between the two different networks.

NOTE: The router R1 can be a Linux-based router, a Juniper Networks device, or any other vendor router.

Figure 2: VPN Topology

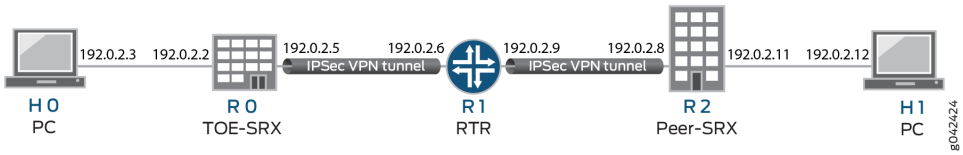


Table 10 on page 74 provides a complete list of the supported IKE protocols, tunnel modes, Phase 1 negotiation mode, authentication method or algorithm, encryption algorithm, DH groups supported for the IKE authentication and encryption (Phase1, IKE Proposal), and for IPsec authentication and encryption (Phase2, IPsec Proposal). The listed protocols, modes, and algorithms are supported and required for 18.1R1 Common Criteria.

Table 10: VPN Combination Matrix

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 1 Proposal (P1, IKE)			
			Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
IKEv1	Main	Route	pre-shared-keys	sha-256	group14	3des-cbc
IKEv2			rsa-signatures-2048	sha-384	group19	aes-128-cbc
			ecdsa-signatures-256		group20	aes-128-gcm
			ecdsa-signatures-384		group24	aes-192-cbc
						aes-256-cbc
						aes-256-gcm

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 2 Proposal (P2, IPsec)			
			Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
IKEv1	Main	Route	hmac-sha1-96	group14	ESP	3des-cbc
IKEv2			hmac-sha-256-128	group19		aes-128-cbc
				group20		aes-128-gcm
				group24		aes-192-cbc
						aes-192-gcm
						aes-256-cbc
						aes-256-gcm

NOTE: The following sections provide sample configurations of IKEv1 IPsec VPN examples for selected algorithms. Authentication and encryption algorithms can be replaced in the configurations to accomplish the user's desired configurations. Use **set security ike gateway <gw-name> version v2-only** command for IKEv2 IPsec VPN.

Configuring an IPsec VPN with a Preshared Key for IKE Authentication

In this section, you configure devices running Junos OS for IPsec VPN using a preshared key as the IKE authentication method. The algorithms used in IKE or IPsec authentication or encryption is shown in Table 11 on page 75

Table 11: IKE or IPsec Authentication and Encryption

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 1 Proposal (P1, IKE)			
			Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
IKEv1	Main	Route	pre-shared-keys	sha-256	group14	aes-256-cbc

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 2 Proposal (P2, IPsec)			
			Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
IKEv1	Main	Route	No Algorithm	group14	ESP	aes-256-cbc

NOTE: A device running Junos OS uses preshared keys for IPsec (no other protocols). TOE accepts ASCII preshared or bit-based keys up to 255 characters (and their binary equivalents) that contain uppercase and lowercase letters, numbers, and special characters such as !, @, #, $, %, ^, &, *, (, and). The device accepts the preshared text keys and converts the text string into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2, using the PRF that is configured as the hash algorithm for the IKE exchanges. The Junos OS does not impose minimum complexity requirements for preshared keys. Hence, users are advised to carefully choose long preshared keys of sufficient complexity.

Configuring IPsec VPN with Preshared Key as IKE Authentication on the Initiator

To configure the IPsec VPN with preshared key IKE authentication on the initiator:

1. Configure the IKE proposal.

```
[edit security ike]
```

```
user@host# set proposal ike-proposal1 authentication-method pre-shared-keys
user@host# set proposal ike-proposal1 dh-group group14
user@host# set proposal ike-proposal1 authentication-algorithm sha256
user@host# set proposal ike-proposal1 encryption-algorithm aes-256-cbc
```

NOTE: Here, ike-proposal1 is the IKE proposal name given by the authorized administrator.

2. Configure the IKE policy.

```
[edit]
user@host# set security ike policy ike-policy1 mode main
user@host# set security ike policy ike-policy1 proposals ike-proposal1
```

NOTE: Here, ike-policy1 is the IKE policy name and ike-proposal1 is the IKE proposal name given by the authorized administrator.

```
user@host# prompt security ike policy ike-policy1 pre-shared-key ascii-text
New ascii-text (secret):
Retype new ascii-text (secret):
```

NOTE: You must enter and reenter the preshared key when prompted. For example, the preshared key can be *CertSqa@jnpr2014*.

NOTE: The preshared key can alternatively be entered in hexadecimal format. For example:

```
[edit]
user@host# prompt security ike policy ike-policy1 hexadecimal
New hexadecimal (secret):
Retype new hexadecimal (secret) (secret):
```

Here, the hexadecimal preshared key can be cc2014bae9876543.

3. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set security proposal ipsec-proposal1 protocol esp
user@host# set security proposal ipsec-proposal1 authentication-algorithm
  hmac-sha-256-128
user@host# set security proposal ipsec-proposal1 encryption-algorithm aes-256-cbc
```

NOTE: Here, ipsec-proposal1 is the IPsec proposal name given by the authorized administrator.

4. Configure the IPsec policy.

```
[edit security ipsec]
user@host# set security policy ipsec-policy1 perfect-forward-secrecy keys group14
user@host# set security policy ipsec-policy1 proposals ipsec-proposal1
```

NOTE: Here, ipsec-policy1 is the IPsec policy name and ipsec-proposal1 is the IPsec proposal name given by the authorized administrator.

5. Configure the IKE.

```
[edit security ike]
user@host# set gateway gw1 ike-policy ike-policy1
user@host# set gateway gw1 address 192.0.2.8
user@host# set gateway gw1 local-identity inet 192.0.2.5
user@host# set gateway gw1 external-interface xe-0/0/2
```

NOTE: Here, gw1 is an IKE gateway name, 192.0.2.8 is the peer VPN endpoint IP, 192.0.2.5 is the local VPN endpoint IP, and xe-0/0/2 is a local outbound interface as the VPN endpoint. The following additional configuration is also needed in the case of IKEv2

```
[edit security ike]
user@host# set gw1 version v2-only
```

6. Configure the VPN.

```
[edit]
user@host# set security ipsec vpn vpn1 ike gateway gw1
user@host# set security ipsec vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# set security ipsec vpn vpn1 bind-interface st0.0
user@host# set routing-options static route 192.0.2.10/24 qualified-next-hop st0.0
  preference 1
```

NOTE: Here, vpn1 is the VPN tunnel name given by the authorized administrator.

7. Configure the outbound flow policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  source-address trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  destination-address untrustLan
```

```
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
  session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then
  session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

8. Configure the inbound flow policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  source-address untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  destination-address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  application any
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then permit
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
  session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
  session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

9. Commit your configuration.

```
user@host# commit
```

Configuring IPsec VPN with Preshared Key as IKE Authentication on the Responder

To configure the IPsec VPN with preshared key IKE authentication on the responder:

1. Configure the IKE proposal.

```
[edit security ike]
user@host# set proposal ike-proposal1 authentication-method pre-shared-keys
user@host# set proposal ike-proposal1 dh-group group14
user@host# set proposal ike-proposal1 authentication-algorithm sha256
user@host# set proposal ike-proposal1 encryption-algorithm 3des-cbc
```

NOTE: Here, ike-proposal1 is the IKE proposal name given by the authorized administrator.

2. Configure the IKE policy.

```
[edit]
user@host# set security ike policy ike-policy1 mode main
user@host# set security ike policy ike-policy1 proposals ike-proposal1
```

NOTE: Here, ike-policy1 is the IKE policy name and ike-proposal1 is the IKE proposal name given by the authorized administrator.

```
user@host# prompt security ike policy ike-policy1 pre-shared-key ascii-text
New ascii-text (secret):
Retype new ascii-text (secret):
```

NOTE: You must enter and reenter the preshared key when prompted. For example, the preshared key can be *CertSqa@jnpr2014*.

NOTE: The pre-share key could alternatively be entered in hexadecimal format. For example,

```
user@host# prompt security ike policy ike-policy1 hexadecimal
New hexadecimal (secret):
Retype new hexadecimal (secret) (secret):
```

Here, the hexadecimal preshared key can be cc2014bae9876543.

3. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set proposal ipsec-proposal1 protocol esp
user@host# set proposal ipsec-proposal1 authentication-algorithm hmac-sha-256-128
user@host# set proposal ipsec-proposal1 encryption-algorithm 3des-cbc
```

NOTE: Here, ipsec-proposal1 is the IPsec proposal name given by the authorized administrator.

4. Configure the IPsec policy.

```
[edit security ipsec]
user@host# set policy ipsec-policy1 perfect-forward-secrecy keys group14
```

```
user@host# set policy ipsec-policy1 proposals ipsec-proposal1
```

NOTE: Here, ipsec-policy1 is the IPsec policy name and ipsec-proposal1 is the IPsec proposal name given by the authorized administrator.

5. Configure the IKE.

```
[edit security ike]
user@host# set gateway gw1 ike-policy ike-policy1
user@host# set gateway gw1 address 192.0.2.5
user@host# set gateway gw1 local-identity inet 192.0.2.8
user@host# set gateway gw1 external-interface xe-0/0/2
```

NOTE: Here, gw1 is an IKE gateway name, 192.0.2.5 is the peer VPN endpoint IP, 192.0.2.8 is the local VPN endpoint IP, and xe-0/0/2 is a local outbound interface as the VPN endpoint. The following additional configuration is also needed in the case of IKEv2.

```
[edit security ike]
user@host# set gw1 version v2-only
```

6. Configure the VPN.

```
[edit]
user@host# set security ipsec vpn vpn1 ike gateway gw1
user@host# set security ipsec vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# set security ipsec vpn vpn1 bind-interface st0.0
user@host# set routing-options static route 192.0.2.7/24 qualified-next-hop st0.0
  preference 1
```

NOTE: Here, vpn1 is the VPN tunnel name given by the authorized administrator.

7. Configure the outbound flow policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  source-address trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  destination-address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
```

```
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
    session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then
    session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

8. Configure the inbound flow policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
    source-address untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
    destination-address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
    application any
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then permit
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
    session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
    session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

9. Commit your configuration.

```
user@host# commit
```

Configuring an IPsec VPN with an RSA Signature for IKE Authentication

The following section provides an example to configure Junos OS devices for IPsec VPN using RSA Signature as IKE Authentication method, whereas, the algorithms used in IKE/IPsec authentication/encryption is as shown in the following table. In this section, you configure devices running Junos OS for IPsec VPN using an RSA signature as the IKE authentication method. The algorithms used in IKE or IPsec authentication or encryption is shown in Table 12 on page 82.

Table 12: IKE/IPsec Authentication and Encryption

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 1 Proposal (P1, IKE)			
			Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
IKEv1	Main	Route	rsa-signatures-2048	sha-256	group14	aes-128-cbc

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 2 Proposal (P2, IPsec)			
			Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
IKEv1	Main	Route	hmac-sha-256-128	group19	ESP	aes-128-cbc

Configuring IPsec VPN with RSA Signature as IKE Authentication on the Initiator

To configure the IPsec VPN with RSA signature IKE authentication on the initiator:

1. Configure the PKI. See Example: Configuring PKI.
2. Generate the RSA key pair. See Example: Generating a Public-Private Key Pair.
3. Generate and load the CA certificate. See Example: Loading CA and Local Certificates Manually.
4. Load the CRL. See Example: Manually Loading a CRL onto the Device .
5. Generate and load a local certificate. See Example: Loading CA and Local Certificates Manually.
6. Configure the IKE proposal.

```
[edit security ike]
user@host# set proposal ike-proposal1 authentication-method rsa-signatures
user@host# set proposal ike-proposal1 dh-group group19
user@host# set proposal ike-proposal1 authentication-algorithm sha-256
user@host# set proposal ike-proposal1 encryption-algorithm aes-128-cbc
```

NOTE: Here, ike-proposal1 is the name given by the authorized administrator.

7. Configure the IKE policy.

```
[edit security ike]
```

```
user@host# set policy ike-policy1 mode main
user@host# set policy ike-policy1 proposals ike-proposal1
user@host# set policy ike-policy1 certificate local-certificate cert1
```

NOTE: Here, ike-policy1 IKE policy name given by the authorized administrator.

8. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set proposal ipsec-proposal1 protocol esp
user@host# set proposal ipsec-proposal1 authentication-algorithm hmac-sha-256-128
user@host# set ipsec-proposal1 encryption-algorithm aes-128-cbc
```

NOTE: Here, ipsec-proposal1 is the name given by the authorized administrator.

9. Configure the IPsec policy.

```
[edit security ipsec]
user@host# set policy ipsec-policy1 perfect-forward-secrecy keys group19
user@host# set policy ipsec-policy1 proposals ipsec-proposal1
```

NOTE: Here, ipsec-policy1 is the name given by the authorized administrator.

10. Configure the IKE.

```
[edit security ike]
user@host# set gateway gw1 ike-policy ike-policy1
user@host# set gateway gw1 address 192.0.2.8
user@host# set gateway gw1 local-identity inet 192.0.2.5
user@host# set gateway gw1 external-interface xe-0/0/1
```

NOTE: Here, 192.0.2.8 is the peer VPN endpoint IP, 192.0.2.5 is the local VPN endpoint IP, and xe-0/0/1 is the local outbound interface as VPN endpoint. The following configuration is also needed for IKEv2.

```
[edit security ike]
user@host# set gw1 version v2-only
```

11. Configure VPN.

```
[edit security ipsec]
user@host# vpn vpn1 ike gateway gw1
user@host# vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# vpn vpn1 bind-interface st0.0
```

NOTE: Here, vpn1 is the VPN tunnel name given by the authorized administrator.

```
user@host# set routing-options static route 192.0.2.10/24 qualified-next-hop st0.0
  preference 1
```

12. Configure the outbound flow policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  source-address trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  destination-address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
  session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
  session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zone and trustLan and untrustLan are preconfigured network addresses.

13. Configure the inbound flow policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  source-address untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  destination-address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  application any
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then permit
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
  session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
  session-close
```

NOTE: Here, **trustZone** and **untrustZone** are preconfigured security zones and **trustLan** and **untrustLan** are preconfigured network addresses.

14. Commit the configuration.

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

Configuring IPsec VPN with RSA Signature as IKE Authentication on the Responder

To configure the IPsec VPN with the RSA signature IKE authentication on the responder:

1. Configure the PKI. See Example: Configuring PKI.
2. Generate the RSA key pair. See Example: Generating a Public-Private Key Pair.
3. Generate and load CA certificate. See Example: Loading CA and Local Certificates Manually.
4. Load the CRL. See Example: Manually Loading a CRL onto the Device .
5. Generate and load a local certificate. See Example: Loading CA and Local Certificates Manually.
6. Configure the IKE proposal.

```
[edit security ike]
user@host# set proposal ike-proposal1 authentication-method rsa-signatures
user@host# set proposal ike-proposal1 dh-group group19
user@host# set proposal ike-proposal1 authentication-algorithm sha-256
user@host# set proposal ike-proposal1 encryption-algorithm aes-128-cbc
```

NOTE: Here, **ike-proposal1** is the name given by the authorized administrator.

7. Configure the IKE policy.

```
[edit security ike]
user@host# set policy ike-policy1 mode main
user@host# set policy ike-policy1 proposals ike-proposal1
user@host# set policy ike-policy1 certificate local-certificate cert1
```

NOTE: Here, ike-policy1 IKE policy name given by the authorized administrator.

8. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set proposal ipsec-proposal1 protocol esp
user@host# set proposal ipsec-proposal1 authentication-algorithm hmac-sha-256-128
user@host# set ipsec-proposal1 encryption-algorithm aes-128-cbc
```

NOTE: Here, ipsec-proposal1 is the name given by the authorized administrator.

9. Configure the IPsec policy.

```
[edit security ipsec]
user@host# set policy ipsec-policy1 perfect-forward-secrecy keys group19
user@host# set policy ipsec-policy1 proposals ipsec-proposal1
```

NOTE: Here, ipsec-policy1 is the name given by the authorized administrator.

10. Configure IKE.

```
[edit security ike]
user@host# set gateway gw1 ike-policy ike-policy1
user@host# set gateway gw1 address 192.0.2.5
user@host# set gateway gw1 local-identity inet 192.0.2.8
user@host# set gateway gw1 external-interface xe-0/0/2
```

NOTE: Here, 192.0.2.5 is the peer VPN endpoint IP, 192.0.2.8 is the local VPN endpoint IP, and xe-0/0/2 is the local outbound interface as VPN endpoint. The following configuration is also needed for IKEv2.

```
[edit security ike]
user@host# set gw1 version v2-only
```

11. Configure VPN.

```
[edit security ipsec]
user@host# vpn vpn1 ike gateway gw1
```

```
user@host# vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# vpn vpn1 bind-interface st0.0
```

NOTE: Here, vpn1 is the VPN tunnel name given by the authorized administrator.

```
user@host# set routing-options static route 192.0.2.1/24 qualified-next-hop st0.0
  preference 1
```

12. Configure the outbound flow policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  source-address trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  destination-address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
  session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
  session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are network addresses.

13. Configure the inbound flow policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  source-address untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  destination-address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  application any
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then permit
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
  session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
  session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

14. Commit the configuration.

    l

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

Configuring an IPsec VPN with an ECDSA Signature for IKE Authentication

In this section, you configure devices running Junos OS for IPsec VPN using an ECDSA signature as the IKE authentication method. The algorithms used in IKE or IPsec authentication or encryption are shown in Table 13 on page 88.

Table 13: IKE or IPsec Authentication and Encryption

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 1 Proposal (P1, IKE)			
			Authentication Method	Authentication Algorithm	DH Group	Encryption Algorithm
IKEv1	Main	Route	ecdsa-signatures-256	sha-384	group14	aes-256-cbc

IKE Protocol	Tunnel Mode	Phase1 Negotiation Mode	Phase 2 Proposal (P2, IPsec)			
			Authentication Algorithm	DH Group (PFS)	Encryption Method	Encryption Algorithm
IKEv1	Main	Route	No Algorithm	group14	ESP	aes-256-gcm

Configuring IPsec VPN with ECDSA signature IKE authentication on the Initiator

To configure the IPsec VPN with ECDSA signature IKE authentication on the initiator:

1. Configure the PKI. See, Example: Configuring PKI.
2. Generate the ECDSA key pair. See Example: Generating a Public-Private Key Pair.
3. Generate and load CA certificate. See Example: Loading CA and Local Certificates Manually.
4. Load CRL. See Example: Manually Loading a CRL onto the Device .
5. Generate and load a local certificate. See Example: Loading CA and Local Certificates Manually.
6. Configure the IKE proposal.

```
[edit security ike]
```

```
user@host# set proposal ike-proposal1 authentication-method ecdsa-signatures-256
user@host# set proposal ike-proposal1 dh-group group14
user@host# set proposal ike-proposal1 authentication-algorithm sha-384
user@host# set proposal ike-proposal1 encryption-algorithm aes-256-cbc
```

NOTE: Here, ike-proposal1 is the IKE proposal name given by the authorized administrator.

7. Configure the IKE policy.

```
[edit security ike]
user@host# set policy ike-policy1 mode main
user@host# set policy ike-policy1 proposals ike-proposal1
user@host# set policy ike-policy1 certificate local-certificate cert1
```

8. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set proposal ipsec-proposal1 protocol esp
user@host# set proposal ipsec-proposal1 encryption-algorithm aes-256-gcm
```

NOTE: Here, ipsec-proposal1 is the IPsec proposal name given by the authorized administrator.

9. Configure the IPsec policy.

```
[edit security ipsec]
user@host# set policy ipsec-policy1 perfect-forward-secrecy keys group14
user@host# set policy ipsec-policy1 proposals ipsec-proposal1
```

NOTE: Here, ipsec-policy1 is the IPsec policy name and ipsec-proposal1 is the IPsec proposal name given by the authorized administrator.

10. Configure IKE.

```
[edit security ike]
user@host# set gateway gw1 ike-policy ike-policy1
user@host# set gateway gw1 address 192.0.2.8
user@host# set gateway gw1 local-identity inet 192.0.2.5
user@host# set gateway gw1 external-interface xe-0/0/2
```

NOTE: Here, gw1 is an IKE gateway name, 192.0.2.8 is the peer VPN endpoint IP, 192.0.2.5 is the local VPN endpoint IP, and xe-0/0/2 is a local outbound interface as the VPN endpoint. The following configuration is also needed for IKEv2.

```
[edit security ike]
user@host# set gw1 version v2-only
```

11. Configure the VPN.

```
[edit]
user@host# set security ipsec vpn vpn1 ike gateway gw1
user@host# set security ipsec vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# set security ipsec vpn vpn1 bind-interface st0.0
user@host# set routing-options static route 192.0.2.10/24 qualified-next-hop st0.0
   preference 1
```

NOTE: Here, vpn1 is the VPN tunnel name given by the authorized administrator.

12. Configure the outbound flow policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
   source-address trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
   destination-address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
   application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
   session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
   session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

13. Configure the inbound flow policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
   source-address untrustLan
```

```
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  destination-address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  application any
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then permit
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
  session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
  session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

14. Commit your configuration.

```
user@host# commit
```

Configuring IPsec VPN with ECDSA signature IKE authentication on the Responder

To configure IPsec VPN with ECDSA signature IKE authentication on the responder:

1. Configure the PKI. See Example: Configuring PKI.
2. Generate the ECDSA key pair. See Example: Generating a Public-Private Key Pair.
3. Generate and load the CA certificate. See Example: Loading CA and Local Certificates Manually.
4. Load the CRL. See Example: Manually Loading a CRL onto the Device.
5. Configure the IKE proposal.

```
[edit security ike]
user@host# set proposal ike-proposal1 authentication-method ecdsa-signatures-256
user@host# set proposal ike-proposal1 dh-group group14
user@host# set proposal ike-proposal1 authentication-algorithm sha-384
user@host# set proposal ike-proposal1 encryption-algorithm aes-256-cbc
```

NOTE: Here, ike-proposal1 is the IKE proposal name given by the authorized administrator.

6. Configure the IKE policy.

```
[edit security ike]
```

```
user@host# set policy ike-policy1 mode main
user@host# set policy ike-policy1 proposals ike-proposal1
user@host# set policy ike-policy1 certificate local-certificate cert1
```

7. Configure the IPsec proposal.

```
[edit security ipsec]
user@host# set proposal ipsec-proposal1 protocol esp
user@host# set proposal ipsec-proposal1 encryption-algorithm aes-256-gcm
```

NOTE: Here, ipsec-proposal1 is the IPsec proposal name given by the authorized administrator.

8. Configure the IPsec policy.

```
[edit security ipsec]
user@host# set policy ipsec-policy1 perfect-forward-secrecy keys group14
user@host# set policy ipsec-policy1 proposals ipsec-proposal1
```

NOTE: Here, ipsec-policy1 is the IPsec policy name and ipsec-proposal1 is the IPsec proposal name given by the authorized administrator.

9. Configure the IKE.

```
[edit security ike]
user@host# set gateway gw1 ike-policy ike-policy1
user@host# set gateway gw1 address 192.0.2.5
user@host# set gateway gw1 local-identity inet 192.0.2.8
user@host# set gateway gw1 external-interface xe-0/0/1
```

NOTE: Here, gw1 is an IKE gateway name, 192.0.2.5 is the peer VPN endpoint IP, 192.0.2.8 is the local VPN endpoint IP, and xe-0/0/1 is a local outbound interface as the VPN endpoint. The following configuration is also needed for IKEv2.

```
[edit security ike]
user@host# set gw1 version v2-only
```

10. Configure the VPN.

```
[edit]
user@host# set security ipsec vpn vpn1 ike gateway gw1
user@host# set security ipsec vpn vpn1 ike ipsec-policy ipsec-policy1
```

```
user@host# set security ipsec vpn vpn1 bind-interface st0.0
user@host# set routing-options static route 192.0.2.1/24 qualified-next-hop st0.0
  preference 1
```

NOTE: Here, vpn1 is the VPN tunnel name given by the authorized administrator.

11. Configure the outbound flow policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  source-address trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  destination-address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
  session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
  session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

12. Configure the inbound flow policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  source-address untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  destination-address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  application any
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then permit
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
  session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
  session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

13. Commit your configuration.

```
user@host# commit
```

Related Documentation

- *Sample Syslog Server Configuration on a Linux System*
- Understanding a Security Flow Policy on a Device Running Junos OS on page 95
- Public Key Infrastructure Feature Guide for Security Devices

CHAPTER 10

Configuring Security Flow Policies

Understanding a Security Flow Policy on a Device Running Junos OS

You can define a security flow policy on a device running Junos OS to inspect and process network packets. The device can permit, deny, and log operations to be associated with each policy. Each of these policies are associated to zones on which distinct network interfaces are bound.

The following modes can be defined for a security flow policy to determine how a device directs traffic:

- Bypass—The **Permit** option directs the traffic traversing the device through the stateful firewall inspection, but not through the IPsec VPN tunnel.
- Discard—The **Deny** option inspects and drops all packets that do not match any **Permit** policies.
- Protect—The traffic is routed through an IPsec tunnel based on the combination of route lookup and **Permit** policy inspection.
- Log—This option logs traffic and session information for all the modes mentioned above.

The following sections describe how to configure a security policy for each of these modes:

Configuring a Security Flow Policy in Firewall Bypass Mode

To configure a security flow policy for firewall bypass mode:

- Configure the security policies.

```
[edit security policies]
```

```
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  source-address trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  destination-address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
  application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
  session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then
  session-close
```

NOTE: Here, **trustZone** and **untrustZone** are preconfigured security zones and **trustLan** and **untrustLan** are preconfigured network addresses. **junos-ssh** is an example of a Junos OS default predefined application that can be configured in a security policy to enforce SSH traffic.

Configuring a Security Policy in Firewall Discard Mode

To configure a security flow policy for firewall discard mode:

- Configure the security policies.

```
[edit security policies]
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  source-address untrustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  destination-address trustLan
user@host# set from-zone untrustZone to-zone trustZone policy policy1 match
  application junos-telnet
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then deny
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then log
  session-init
user@host# set from-zone untrustZone to-zone trustZone policy policy1 then
  session-close
```

NOTE: Here, **trustZone** and **untrustZone** are the preconfigured security zones and **trustLan** and **untrustLan** are preconfigured network addresses. **junos-telnet** is an example of a Junos OS default predefined application that can be configured in a security policy to enforce Telnet traffic.

Configuring a Security Flow Policy in IPsec Protect Mode

To configure a security flow policy for IPSec protect mode:

1. Configure the VPN.

```
[edit]
user@host# set security ipsec vpn vpn1 ike gateway gw1
user@host# set security ipsec vpn vpn1 ike ipsec-policy ipsec-policy1
user@host# set security ipsec vpn vpn1 bind-interface st0.0
user@host# set routing-options static route 198.51.100.14/24 qualified-next-hop st0.0
    preference 1
```

NOTE: Here, gw1 and ipsec-policy1 are preconfigured IKE and IPsec policies.

2. Configure the security policies.

```
[edit security policies]
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
    source-address trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
    destination-address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
    application any
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
    session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then
    session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

Related Documentation

- Configuring VPN on a Device Running Junos OS on page 73

CHAPTER 11

Configuring Traffic Filtering Rules

Understanding Protocol Support

You can configure the devices running Junos OS to perform stateful network traffic filtering on network packets using network traffic protocols and network fields as described in Table 7 on page 67.

Table 14: Network Traffic Protocols and Fields

Protocol or RFC	Fields
ICMPv4 - RFC 792, Internet Control Message Protocol version 4	• Type • Code
ICMPv6 - RFC 4443, Internet Control Message Protocol version 6	• Type • Code
IPv4 - RFC 791, Internet Protocol	• Source address • Destination address • Transport Layer Protocol
IPv6 - RFC 2460, Internet Protocol	• Source address • Destination address • Transport Layer Protocol

Table 14: Network Traffic Protocols and Fields (continued)

Protocol or RFC	Fields
TCP - RFC 793, Transmission Control Protocol	• Source port • Destination port
UDP - RFC 768, User Datagram Protocol	• Source port • Destination port

The following protocols are also supported on devices running Junos OS and are a part of this evaluation.

- IPsec
- IKE
- SSH

The following protocols are supported on devices running Junos OS but are not included in the scope of this evaluation.

- OSPF
- BGP
- RIP

NOTE: SSH is not evaluated on devices running Junos OS but is provided for remote administration, contingent on SSH through IPsec being used for connecting the device.

Related Documentation

- Configuring Traffic Filter Rules on page 100

Configuring Traffic Filter Rules

Traffic filter rules can be configured on a device to enforce validation against protocols attributes and direct traffic accordingly to the configured attributes. These rules are based on zones on which network interfaces are bound.

The following procedure describes how to configure traffic filter rules to direct FTP traffic from source **trustZone** to destination **untrustZone** and from source network **trustLan** to destination network **untrustLan**. Here, traffic is traversing from the devices interface A on **trustZone** to interface B on **untrustZone**.

1. Configure a zone and its interfaces.

```
[edit]
user@host# set security zones security-zone trustLan interfaces xe-0/0/0
```

2. Configure the security policy in the specified zone-to-zone direction and specify the match criteria.

```
[edit security policies
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
   source-address trustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
   destination-address untrustLan
user@host# set from-zone trustZone to-zone untrustZone policy policy1 match
   application ftp
```

3. Configure the security policy in the specified zone-to-zone direction and specify the action to take when a packet matches a criteria.

```
[edit security policies
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then permit
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then log
   session-init
user@host# set from-zone trustZone to-zone untrustZone policy policy1 then
   session-close
```

NOTE: Here, trustZone and untrustZone are preconfigured security zones and trustLan and untrustLan are preconfigured network addresses.

Related Documentation

- Understanding Protocol Support on page 99

Configuring Default Deny-All and Reject Rules

By default, security devices running Junos OS deny traffic unless rules are explicitly created to allow it using the following command:

```
[edit]
user@host#set security policies default-policy deny-all
```

You can configure your security devices running Junos OS to enforce the following default reject rules with logging on all network traffic:

- Invalid fragments
- Fragmented IP packets that cannot be reassembled completely
- Where the source address is equal to the address of the network interface
- Where the source address does not belong to the networks associated with the network interface
- Where the source address is defined as being on a broadcast network
- Where the source address is defined as being on a multicast network

- Where the source address is defined as being a loopback address
- Where the source address is a multicast packet
- Where the source or destination address is a link-local address
- Where the source or destination address is defined as being an address “reserved for future use” as specified in RFC 5735 for IPv4
- Where the source or destination address is defined as an “unspecified address” or an address “reserved for future definition and use” as specified in RFC 3513 for IPv6
- With the IP option Loose Source Routing, Strict Source Routing, or Record Route is specified

Logging the Dropped Packets Using Default Deny-all Option

The evaluated configuration device drops all IPv6 traffic by default. This topic describes how to log packets dropped by this default deny-all option.

Before you begin, log in with your root account on a Junos OS device running Junos OS Release 18.1R1 and edit the configuration.

NOTE: You can enter the configuration commands in any order and commit all the commands at once.

To log packets dropped by the default deny-all option:

1. Configure a network security policy in a global context and specify the security policy match criteria.

```
[edit security policy]
user@host# set global policy always-last-default-deny-and-log match source-address
   any destination-address any application any
```

2. Specify the policy action to take when the packet matches the criteria.

```
[edit security policy]
user@host# set global policy always-last-default-deny-and-log then deny
```

3. Configure the security policy to enable logs at the session initialization time.

```
[edit security policy]
user@host# set global policy always-last-default-deny-and-log then log session-init
```

NOTE: This procedure might capture a very large amount of data until you have configured the other policies.

To permit all IPv6 traffic into an SRX Series device, configure the device with flow-based forwarding mode. While the default policy in flow-based forwarding mode is still to drop all IPv6 traffic, you can now add rules to permit selected types of IPv6 traffic.

```
user@host# set security forwarding-options family inet6 mode flow-based
```

Configuring Mandatory Reject Rules for Invalid Fragments and Fragmented IP Packets

This topic describes how to configure mandatory reject rules for invalid fragments and fragmented IP packets that cannot be reassembled.

Before you begin, log in with your root account on a Junos OS device running Junos OS Release 18.1R1 and edit the configuration.

NOTE: You can enter the configuration commands in any order and commit all the commands at once.

To configure mandatory reject rules:

1. Specify the flow configuration to forcefully reassemble the IP fragments.

```
[edit]
user@host# set security flow force-ip-reassembly
```

2. Delete the screen ID and the IDS options and enable the ICMP fragment IDS option.

```
[edit]
user@host# delete security screen ids-option trustScreen icmp fragment
```

3. Delete the IP layer IDS option and enable the IP fragment blocking IDS option.

```
[edit]
user@host# delete security screen ids-option trustScreen ip block-frag
```

Configuring Default Reject Rules for Source Address Spoofing

The following guidelines describe when to configure the default reject rules for source address spoofing:

- When the source address is equal to the address of the network interface where the network packet was received.
- When the source address does not belong to the networks associated with the network interface where the network packet was received.
- When the source address is defined as being on a broadcast network.

Before you begin, log in with your root account on a Junos OS device running Junos OS Release 18.1R1 and edit the configuration.

NOTE: You can enter the configuration commands in any order and commit all the commands at once.

To configure default reject rules to log source address spoofing:

1. Configure the security screen features and enable the IP address spoofing IDS option.

```
[edit]
user@host# set security screen ids-option trustScreen ip spoofing
```

2. Specify the name of the security zone and the IDS option object applied to the zone.

```
[edit]
user@host# set security zones security-zone trustZone screen trustScreen
```

Configuring Default Reject Rules with IP Options

This topic describes how to configure default reject rules with IP options. The IP options enable the device to either block any packets with loose or strict source route options or detect such packets and then record the event in the counters list for the ingress interface.

Before you begin, log in with your root account to an SRX Series device running Junos OS Release 18.1R1.

NOTE: You can enter the configuration commands in any order and commit all the commands at once.

To configure the default reject rules with IP options:

1. Configure the screen features to enable IP options.

```
[edit security screen ids-option trustScreen]
user@host# set ip source-route-option
user@host# set ip loose-source-route-option
user@host# set ip strict-source-route-option
user@host# set ip record-route-option
```

2. Specify the name of the security zone and the IDS option object applied to the zone.

```
[edit]
user@host# set security zones security-zone trustZone screen trustScreen
```

Configuring Default Reject Rules

The following guidelines describe when to configure the default reject rules:

- Source address is defined on a multicast network, a loopback address, or a multicast address.
- The source or destination address of a packet is a link-local address, an address "reserved for future use" as specified in RFC 5735 for IPv4, an "unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6.
- An illegal or out-of-sequence TCP packet is received.

Before you begin, log in with your root account on a Junos OS device running Junos OS Release 18.1R1 and edit the configuration.

NOTE: You can enter the configuration commands in any order and commit all the commands at once.

To configure default reject rules:

1. Configure the security screen features and enable the IP address spoofing IDS option.

```
[edit security]
user@host# set security screen ids-option trustScreen ip spoofing
```

2. Configure the security flow feature to log the dropped illegal packets.

```
[edit security]
user@host# set security flow log dropped-illegal-packet
```

3. Specify the name of the security zone and the IDS option object applied to the zone.

```
[edit security]
user@host# set security zones security-zone trustZone screen trustScreen
```

4. Configure the mandatory TCP reject rule.

```
[edit security]
user@host# set security flow tcp-session strict-syn-check
```

Configuring the Device to Drop Unassigned IPv6 Packets

Before you configure the device to drop unassigned IPv6 packets, check the default configuration status of the device. From the operational mode enter the **show usp flow** configuration command.

In the output, the advanced options **no_drop_unassigned_ipv6_address: disabled (default)** option indicates that by default the device drops unassigned IPv6 packets.

To enable the device to drop unassigned IPv6 packets, use the following command:

```
user@host# set security flow advanced-options no-drop-unassigned-ipv6-address
user@host# commit
```

To enable users to revert to the default configuration, use the following command:

```
user@host# delete security flow advanced-options no-drop-unassigned-ipv6-address
user@host# commit
```

CHAPTER 12

Configuring Network Attacks

Configuring IP Teardrop Attack Screen

This topic describes how to configure detection of an IP teardrop attack.

Teardrop attacks exploit the reassembly of fragmented IP packets. In the IP header, one of the field is the fragment offset fields, which indicates the starting position, or offset of the data contained in a fragmented packet, relative to the data of the original unfragmented packet. When the sum of the offset and size of one fragmented packet differs from that of the next fragmented packet, the packets overlap and the server attempting to reassemble the packet might crash.

To enable detection of a teardrop attack:

1. Configure interfaces and assign IP addresses to the interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure the security screen option and attach it to the **untrustZone**.

```
user@host# set security screen ids-option untrustScreen ip tear-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring TCP Land Attack Screen

This topic describes how to configure detection of a TCP land attack.

Land attacks occur when an attacker sends spoofed SYN packets containing the IP address of the victim as both the destination and the source IP address.

To enable detection of a TCP land attack:

1. Configure interfaces and assign IP addresses to the interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
user@host# set security screen ids-option untrustScreen tcp land
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
```

```
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring ICMP Fragment Screen

This topic describes how to configure detection of an ICMP fragment attack.

If an ICMP packet is large, then it must be fragmented. When the ICMP fragment protection screen option is enabled, the Junos OS blocks any ICMP packet that has many fragment flags set or that has an offset value indicated in the offset field.

To enable detection of an ICMP fragment IDS attack:

1. Configure interfaces and assign an IP address to interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
```

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
user@host# set security screen ids-option untrustScreen icmp fragment
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring Ping-Of-Death Attack Screen

This topic describes how to configure detection of ping-of-death attack.

The IP datagram with the protocol field of the IP header is set to 1 (ICMP), the last fragment bit is set, and (IP offset * 8) + (IP data length) > 65535. The IP offset (which represents the starting position of this fragment in the original packet, and which is in 8-byte units) plus the rest of the packet is greater than the maximum size for an IP packet.

To enable detection of a ping-of-death IDP attack:

1. Configure interfaces and assign an IP address to interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
```

```
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
user@host# set security screen ids-option untrustScreen icmp ping-death
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring tcp-no-flag Attack Screen

This topic describes how to configure detection of a **tcp-no-flag** attack.

A TCP segment with no control flags set is an anomalous event causing various responses from the recipient. When the TCP no-flag is enabled, the device detects the TCP segment headers with no flags set, and drops all TCP packets with missing or malformed flag fields.

To enable detection of a **tcp-no-flag** option:

1. Configure interfaces and assign an IP address to the interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
user@host# set security screen ids-option untrustScreen tcp tcp-no-flag
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring TCP SYN-FIN Attack Screen

This topic describes how to configure detection of a TCP SYN-FIN attack.

A TCP header with the SYN and FIN flags set is anomalous TCP behavior causing various responses from the recipient, depending on the OS. Blocking packets with SYN and FIN flags helps prevent the OS system probes.

To enable detection of TCP SYN-FIN bits:

1. Configure interfaces and assign an IP address to interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
```

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
user@host# set security screen ids-option untrustScreen tcp syn-fin
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring TCP fin-no-ack Attack Screen

This topic describes how to configure detection of TCP **fin-no-ack** attack. A TCP header with the FIN flag set but not the ACK flag is anomalous TCP behavior.

To enable detection of FIN bits with no ACK bit IDS option:

1. Configure interfaces and assign an IP address to interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
```

```
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
user@host# set security screen ids-option untrustScreen tcp fin-no-ack
user@host# set security zones security-zone untrustZone screen untrustScreen
user@host# set security screen ids-option untrustScreen alarm-without-drop
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring UDP Bomb Attack Screen

If the UDP length specified is less than the IP length specified then the malformed packet type is associated with a denial-of-service attempt. By default, SRX drops these packets. No configuration is required.

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring UDP CHARGEN DoS Attack Screen

This topic describes how to configure protection from a UDP CHARGEN DoS attack.

NOTE: UDP packet is detected with a source port of 7 and a destination port of 19 is an attack.

To enable detection of a UDP CHARGEN DoS attack:

1. Configure interfaces and assign an IP address to interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to the **trustZone** with the Junos OS predefined application **junos-chargen**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application junos-chargen
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then deny
user@host# set security policies default-policy permit-all
```

4. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

5. To allow the packet to reach the destination, change the policy configuration from **deny** to **permit**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring TCP SYN and RST Attack Screen

This topic describes how to configure TCP packet with the SYN and RST flags are set.

To configure the TCP SYN and RST attack:

1. Configure interfaces and assign IP address to interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 1.1.1.1/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 2.2.2.1/24
```

2. Configure security zones “trustZone” and “untrustZone” and assign interfaces.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure IDP custom-attack signatures.

```
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match from-zone
  any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match
  source-address any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match to-zone
  any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match
  destination-address any
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match application
  default
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 match attacks
  custom-attacks syn_rst
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 then action
  no-action
user@host# set security idp idp-policy idpengine rulebase-ips rule 1 then notification
  log-attacks
user@host# set security idp active-policy idpengine
user@host# set security idp custom-attack syn_rst severity info
user@host# set security idp custom-attack syn_rst attack-type signature context
  packet
user@host# set security idp custom-attack syn_rst attack-type signature pattern
user@host# set security idp custom-attack syn_rst attack-type signature direction
  any
user@host# set security idp custom-attack syn_rst attack-type signature protocol tcp
  tcp-flags rst
user@host# set security idp custom-attack syn_rst attack-type signature protocol tcp
  tcp-flags syn
```

4. Configure security policies from "untrustZone" to "trustZone".

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit application-services idp
user@host# set security policies default-policy deny-all
```

5. Configure security tcp-session options in flow.

```
user@host# set security flow tcp-session no-syn-check
user@host# set security flow tcp-session no-sequence-check
```

6. Configuring syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
```

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

7. To allow the traffic to reach destination, configure below **tcp-session** option.

```
user@host# set security flow tcp-session relax-check
```

Related Documentation

-

Configuring ICMP Flood Attack Screen

This topic describes how to configure detection of an ICMP flood attack.

An ICMP flood typically occurs when an ICMP echo request overloads the victim with many requests such that the ICMP echo request spends all its resources responding until it can no longer process valid network traffic. When enabling the ICMP flood protection feature, you can set a threshold that, once exceeded, invokes the ICMP flood attack protection feature.

To enable detection of an ICMP flood attack:

1. Configure interfaces and assign an IP address to interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
```

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
user@host# set security screen ids-option untrustScreen icmp flood
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring TCP SYN Flood Attack Screen

This topic describes how to configure detection of a TCP SYN flood attack.

A SYN flood occurs when a host is so overwhelmed by SYN segments initiating incomplete connection requests that it can no longer process legitimate connection requests.

To enable detection of a TCP SYN flood attack:

1. Configure interfaces and assign an IP address to interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
```

```
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
user@host# set security screen ids-option untrustScreen tcp syn-flood
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring TCP Port Scan Attack Screen

This topic describes how to configure detection of a TCP port scan attack.

A port scan occurs when one source IP address sends an IP packet containing TCP SYN segments to a defined number of different ports at the same destination IP address within a defined interval.

To enable detection of a TCP port scan attack:

1. Configure interfaces and assign an IP address to interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
user@host# set security screen ids-option untrustScreen tcp port-scan
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
```

```
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring UDP Port Scan Attack Screen

This topic describes how to configure detection of a UDP port scan attack.

These attacks scan the target IP addresses for open, listening, or responsive services by targeting multiple protocols or ports on one or more target IP address using obvious (sequentially numbered) patterns of the target protocol or port numbers. The patterns are derived by randomizing the protocol or port numbers and randomizing the time delays between the transmissions.

To enable detection of a UDP port scan attack:

1. Configure interfaces and assign an IP address to interfaces.

```
user@host# set interfaces xe-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces xe-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces xe-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces xe-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
```

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
user@host# set security screen ids-option untrustScreen udp port-scan
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

Configuring IP Sweep Attack Screen

This topic describes how to configure detection of an IP sweep attack.

An address sweep occurs when one source IP address sends a defined number of ICMP packets to different hosts within a defined time interval (5000 microseconds is the default value). The purpose of this attack is to send ICMP packets—typically echo requests—to various hosts in the hope that at least one replies, thus uncovering an address to target.

To enable detection of an IP sweep attack:

1. Configure interfaces and assign an IP address to interfaces.

```
user@host# set interfaces ge-0/0/1 unit 0 family inet address 192.0.2.0/24
user@host# set interfaces ge-0/0/3 unit 0 family inet address 198.51.100.0/24
```

2. Configure security zones **trustZone** and **untrustZone** and assign interfaces to them.

```
user@host# set security zones security-zone trustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone trustZone host-inbound-traffic protocols
  all
user@host# set security zones security-zone trustZone interfaces ge-0/0/1.0
user@host# set security zones security-zone untrustZone host-inbound-traffic
  system-services all
user@host# set security zones security-zone untrustZone host-inbound-traffic
  protocols all
user@host# set security zones security-zone untrustZone interfaces ge-0/0/3.0
```

3. Configure security policies from **untrustZone** to **trustZone**.

```
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match source-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match destination-address any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 match application any
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then permit
user@host# set security policies default-policy deny-all
```

4. Configure security screens and attach them to **untrustZone**.

```
user@host# set security screen ids-option untrustScreen icmp ip-sweep
user@host# set security screen ids-option untrustScreen alarm-without-drop
user@host# set security zones security-zone untrustZone screen untrustScreen
```

5. Configure syslog.

```
user@host# set system syslog file syslog any any
user@host# set system syslog file syslog archive size 10000000
user@host# set system syslog file syslog explicit-priority
user@host# set system syslog file syslog structured-data
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-init
user@host# set security policies from-zone untrustZone to-zone trustZone policy
  policy1 then log session-close
```

6. Commit the configuration.

```
user@host# commit
```

Related Documentation

- IDP Extended Package Configuration Overview on page 127

CHAPTER 13

Configuring the IDP Extended Package

IDP Extended Package Configuration Overview

The Junos OS Intrusion Detection and Prevention (IDP) policy enables you to selectively enforce various attack detection and prevention techniques on network traffic passing through an IDP-enabled device. It allows you to define policy rules to match a section of traffic based on a zone, network, and application, and then take active or passive preventive actions on that traffic.

An IDP policy defines how your device handles the network traffic. It allows you to enforce various attack detection and prevention techniques on traffic traversing your network.

A policy is made up of rule bases, and each rule base contains a set of rules. You define rule parameters, such as traffic match conditions, action, and logging requirements, then add the rules to rule bases. After you create an IDP policy by adding rules in one or more rule bases, you can select that policy to be the active policy on your device.

To configure the IDP extended package (IPS-EP) perform the following steps:

1. Enable IPS in a security policy. See *Configuring IDP Policy Rules and IDP Rulebases* in the Junos OS Release 12.3X48-D10 Intrusion Detection and Prevention Feature Guide for Security Devices published on 2016-01-12 for Junos OS Release 12.3X48-D10 released on 2015-03-06.

2. Configure IDP policy rules, IDP rule bases, and IDP rule actions. See *Configuring IDP Policy Rules and IDP Rulebases* in the Junos OS Release 12.3X48-D10 Intrusion Detection and Prevention Feature Guide for Security Devices published on 2016-01-12 for Junos OS Release 12.3X48-D10 released on 2015-03-06.

3. Configure IDP custom signatures. See *Understanding IDP Signature-Based Attacks* and *Example: Configuring IDP Signature-Based Attacks* in the Junos OS Release 12.3X48-D10 Intrusion Detection and Prevention Feature Guide for Security Devices published on 2016-01-12 for Junos OS Release 12.3X48-D10 released on 2015-03-06.

4. Update the IDP signature database. See *Updating the IDP Signature Database Overview* in the Junos OS Release 12.3X48-D10 Intrusion Detection and Prevention Feature Guide

for Security Devices published on 2016-01-12 for Junos OS Release 12.3X48-D10 released on 2015-03-06.

Related Documentation

- Intrusion Detection and Prevention Feature Guide for Security Devices

CHAPTER 14

Performing Self-Tests on a Device

Understanding FIPS Self-Tests

The cryptographic module enforces security rules to ensure that a device running the Juniper Networks Junos operating system (Junos OS) in FIPS mode of operation meets the security requirements of FIPS 140-2 Level 2. To validate the output of cryptographic algorithms approved for FIPS and test the integrity of some system modules, the device performs the following series of known answer test (KAT) self-tests:

- **md_kats**—KAT for libmd and libc
- **quicksec_7_0_kats**—KAT for Quicksec_7_0 Toolkit cryptographic implementation
- **openssl_kats**—KAT for OpenSSL cryptographic implementation
- **kernel_kats**——KAT for kernel cryptographic routines
- **srxpfe_kats**—KAT for SRX packet forwarding engine

The KAT self-tests are performed automatically at startup and reboot, regardless of whether FIPS mode of operation is enabled on the device. Conditional self-tests are also performed automatically to verify digitally signed software packages, generated random numbers, RSA and ECDSA key pairs, and manually entered keys.

If the KATs are completed successfully, the system log (syslog) file is updated to display the tests that were executed.

If the device fails a KAT, the device writes the details to a system log file, enters FIPS error state (panic), and reboots.

The **file show /var/log/messages** command displays the system log.

Performing Power-On Self-Tests on the Device

Each time the cryptographic module is powered on, the module tests that the cryptographic algorithms still operate correctly and that sensitive data has not been damaged. Power-on self-tests are performed on demand by power cycling the module.

On powering on or resetting the device, the module performs the following self-tests. All KATs must be completed successfully prior to any other use of cryptography by the module. If one of the KATs fail, the module enters the Critical Failure error state.

The module displays the following status output for SRX4600 device while running the power-on self-tests:

```
root@host:fips> request system fips self-test
Testing kernel KATS:
  NIST 800-90 HMAC DRBG Known Answer Test:         Passed
  DES3-CBC Known Answer Test:                      Passed
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  SHA-2-384 Known Answer Test:                     Passed
  SHA-2-512 Known Answer Test:                     Passed
  AES128-CMAC Known Answer Test:                   Passed
  AES-CBC Known Answer Test:                       Passed
Testing MacSec KATS:
  AES128-CMAC Known Answer Test:                   Passed
  AES256-CMAC Known Answer Test:                   Passed
  AES-KEYWRAP Known Answer Test:                   Passed
Testing libmd KATS:
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  SHA-2-512 Known Answer Test:                     Passed
Testing OpenSSL KATS:
  FIPS RNG Known Answer Test:                      Passed
  NIST 800-90 HMAC DRBG Known Answer Test:         Passed
  FIPS ECDSA Known Answer Test:                    Passed
  FIPS ECDH Known Answer Test:                     Passed
  FIPS RSA Known Answer Test:                      Passed
  DES3-CBC Known Answer Test:                      Passed
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-224 Known Answer Test:                 Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  HMAC-SHA2-384 Known Answer Test:                 Passed
  HMAC-SHA2-512 Known Answer Test:                 Passed
  AES-CBC Known Answer Test:                       Passed
  AES-GCM Known Answer Test:                       Passed
  ECDSA-SIGN Known Answer Test:                    Passed
  KDF-IKE-V1 Known Answer Test:                    Passed
  KDF-SSH-SHA256 Known Answer Test:                Passed
Testing QuickSec 7.0 KATS:
  NIST 800-90 HMAC DRBG Known Answer Test:         Passed
  DES3-CBC Known Answer Test:                      Passed
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-224 Known Answer Test:                 Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  HMAC-SHA2-384 Known Answer Test:                 Passed
  HMAC-SHA2-512 Known Answer Test:                 Passed
  AES-CBC Known Answer Test:                       Passed
  AES-GCM Known Answer Test:                       Passed
  SSH-RSA-ENC Known Answer Test:                   Passed
  SSH-RSA-SIGN Known Answer Test:                  Passed
  SSH-ECDSA-SIGN Known Answer Test:                Passed
  KDF-IKE-V1 Known Answer Test:                    Passed
  KDF-IKE-V2 Known Answer Test:                    Passed
Testing QuickSec KATS:
```

```
  NIST 800-90 HMAC DRBG Known Answer Test:         Passed
  DES3-CBC Known Answer Test:                      Passed
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-224 Known Answer Test:                 Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  HMAC-SHA2-384 Known Answer Test:                 Passed
  HMAC-SHA2-512 Known Answer Test:                 Passed
  AES-CBC Known Answer Test:                       Passed
  AES-GCM Known Answer Test:                       Passed
  SSH-RSA-ENC Known Answer Test:                   Passed
  SSH-RSA-SIGN Known Answer Test:                  Passed
  KDF-IKE-V1 Known Answer Test:                    Passed
  KDF-IKE-V2 Known Answer Test:                    Passed
Testing SSH IPsec KATS:
  NIST 800-90 HMAC DRBG Known Answer Test:         Passed
  DES3-CBC Known Answer Test:                      Passed
  HMAC-SHA1 Known Answer Test:                     Passed
  HMAC-SHA2-256 Known Answer Test:                 Passed
  AES-CBC Known Answer Test:                       Passed
  SSH-RSA-ENC Known Answer Test:                   Passed
  SSH-RSA-SIGN Known Answer Test:                  Passed
  KDF-IKE-V1 Known Answer Test:                    Passed
Testing file integrity:
  File integrity Known Answer Test:                Passed
Testing crypto integrity:
  Crypto integrity Known Answer Test:              Passed
Expect an exec Authentication error...
/sbin/kats/run-tests: /sbin/kats/cannot-exec: Authentication error
```

NOTE: The module implements cryptographic libraries and algorithms that are not utilized in the approved mode of operation.

Related Documentation

- How to Enable and Configure Junos OS in FIPS Mode of Operation on page 38

CHAPTER 15

Configuration Statements

checksum-validate

Syntax

```
checksum-validate {
  match (equal | greater-than | less-than | not-equal);
  value checksum-value;
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol ipv4]
[edit security idp custom-attack attack-name attack-type signature protocol tcp]
[edit security idp custom-attack attack-name attack-type signature protocol udp]
[edit security idp custom-attack attack-name attack-type signature protocol icmp]
[edit security idp custom-attack attack-name attack-type signature protocol icmpv6]
```

Release Information Statement introduced in Junos OS Release 12.3X48-D30.

Description Allow IDP to validate checksum field against the calculated checksum.

Options **match (equal | greater-than | less-than | not-equal)**—Match an operand.

value *checksum-value*—Match a decimal value.

Range: 0 through 65,535

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

code

Syntax

```
code {
  match (equal | greater-than | less-than | not-equal);
  value code-value;
}
```

Hierarchy Level

[edit security idp custom-attack *attack-name* attack-type signature protocol icmpv6]

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

Description

Specify the secondary code that identifies the function of the request/reply within a given type.

Options

- **match** (**equal** | **greater-than** | **less-than** | **not-equal**)—Match an operand.
- **value** ***code-value***—Match a decimal value.

  Range: 0 through 255

Required Privilege Level

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

data-length

Syntax

```
data-length {
  match (equal | greater-than | less-than | not-equal);
  value data-length;
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol udp]
[edit security idp custom-attack attack-name attack-type signature protocol icmp]
[edit security idp custom-attack attack-name attack-type signature protocol icmpv6]
[edit security idp custom-attack attack-name attack-type signature protocol tcp]
```

Release Information Statement introduced in Junos OS Release 9.3.

Description Specify the number of bytes in the data payload. In the TCP header, for SYN, ACK, and FIN packets, this field should be empty.

Options

- **match** (**equal** | **greater-than** | **less-than** | **not-equal**)—Match an operand.
- **value** ***data-length***—Match the number of bytes in the data payload.

  Range: 0 through 65,535

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

destination-option

Syntax

```
destination-option {
  home-address {
    match (equal | greater-than | less-than | not-equal);
    value header-value;
  }
  option-type {
    match (equal | greater-than | less-than | not-equal);
    value header-value;
  }
}
```

Hierarchy Level

```
[edit set security idp custom-attack attack-name attack-type signature protocol ipv6
  extension-header]
```

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

Description

Specify the IPv6 destination option for the extension header. The **destination-option** option inspects the header option type of **home-address** field in the **extension header** and reports a custom attack if a match is found. The **destination-option** supports the **home-address** field type of inspection.

Required Privilege Level

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

extension-header

Syntax

```
extension-header {
  destination-option {
    home-address {
      match (equal | greater-than | less-than | not-equal);
      value header-value;
    }
      option-type {
        match (equal | greater-than | less-than | not-equal);
        value header-value;
      }
  }
  routing-header {
    header-type {
      match (equal | greater-than | less-than | not-equal);
      value header-value;
      }
    }
}
```

Hierarchy Level [edit set security idp custom-attack *attack-name* attack-type signature protocol *ipv6*]

Release Information Statement introduced in Junos OS Release 12.3X48-D30.

Description Specify the IPv6 extension header.

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

header-type

Syntax

```
header-type {
    match (equal | greater-than | less-than | not-equal);
    value header-value;
}
```

Hierarchy Level

```
[edit set security idp custom-attack attack-name attack-type signature protocol ipv6
    extension-header routing-header]
```

Release Information Statement introduced in Junos OS Release 12.3X48-D30.

Description Specify the IPv6 routing header type.

Options **match (equal | greater-than | less-than | not-equal)**—Match an operand.

value—Match a decimal value.

Range: 0 through 255

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

home-address

Syntax

```
home-address {
  match (equal | greater-than | less-than | not-equal);
  value value;
}
```

Hierarchy Level

```
[edit set security idp custom-attack attack-name attack-type signature protocol ipv6
  extension-header destination-option]
```

Release Information Statement introduced in Junos OS Release 12.3X48-D30.

Description Specify the IPv6 home address of the mobile node.

Options **match (equal | greater-than | less-than | not-equal)**—Match an operand.

value—Match a decimal value.

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

identification

Syntax

```
identification {
  match (equal | greater-than | less-than | not-equal);
  value identification-value;
}
```

Hierarchy Level

[edit security idp custom-attack *attack-name* attack-type signature protocol icmpv6]

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

Description

Specify a unique value used by the destination system to associate requests and replies.

Options

- **match** (**equal** | **greater-than** | **less-than** | **not-equal**)—Match an operand.
- **value** ***identification-value***—Match a decimal value.

  Range: 0 through 65,535

Required Privilege Level

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

icmpv6 (Security IDP Custom Attack)

Syntax

```
icmpv6 {
  checksum-validate {
    match (equal | greater-than | less-than | not-equal);
    value checksum-value;
  }
  code {
    match (equal | greater-than | less-than | not-equal);
    value code-value;
  }
  data-length {
    match (equal | greater-than | less-than | not-equal);
    value data-length;
  }
  identification {
    match (equal | greater-than | less-than | not-equal);
    value identification-value;
  }
  sequence-number {
    match (equal | greater-than | less-than | not-equal);
    value sequence-number;
  }
  type {
    match (equal | greater-than | less-than | not-equal);
    value type-value;
  }
}
```

Hierarchy Level

[edit security idp custom-attack *attack-name* attack-type signature protocol]

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

Description

Allow IDP to match the attack for the specified ICMPv6.

Required Privilege Level

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

ihl (Security IDP Custom Attack)

Syntax

```
ihl {
  match (equal | greater-than | less-than | not-equal);
  value ihl-value;
}
```

Hierarchy Level

[edit set security idp custom-attack *ipv4_custom* attack-type signature protocol *ipv4*]

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

Description

Specify the IPv4 header length in words.

Options

match (equal | greater-than | less-than | not-equal)—Match an operand.

value—Match a decimal value.

Range: 0 through 15

Required Privilege Level

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

option-type

Syntax

```
option-type {
  match (equal | greater-than | less-than | not-equal);
  value header-value;
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol ipv6
  extension-header destination-option]
```

Release Information Statement introduced in Junos OS Release 12.3X48-D30.

Description Specify the type of option for destination header type.

Options **match (equal | greater-than | less-than | not-equal)**—Match an operand.

value—Match a decimal value.

Range: 0 through 255

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

reserved (Security IDP Custom Attack)

Syntax

```
reserved {
  match (equal | greater-than | less-than | not-equal);
  value reserved-value;
}
```

Hierarchy Level

[edit security idp custom-attack *ipv4_custom* attack-type signature protocol *tcp*]

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

Description

Specify the three reserved bits in the TCP header field.

Options

match (equal | greater-than | less-than | not-equal)—Match an operand.

value—Match a decimal value.

Range: 0 through 7

Required Privilege Level

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

routing-header

Syntax

```
routing-header {
  header-type {
    match (equal | greater-than | less-than | not-equal);
    value header-value;
  }
}
```

Hierarchy Level

```
[edit set security idp custom-attack attack-name attack-type signature protocol ipv6
  extension-header]
```

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

Description

Specify the IPv6 routing header type. The **routing-header** option inspects the routing-header type field and reports a custom attack if a match with the specified value is found. The **routing-header** option supports the following routing header types: **routing-header-type0**, **routing-header-type1**, and so on.

Required Privilege Level

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

sequence-number (Security IDP ICMPv6 Headers)

Syntax

```
sequence-number {
  match (equal | greater-than | less-than | not-equal);
  value sequence-number;
}
```

Hierarchy Level

[edit security idp custom-attack *attack-name* attack-type signature protocol icmpv6]

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

Description

Specify the sequence number of the packet. This number identifies the location of the request/reply in relation to the entire sequence.

Options

- **match** (**equal** | **greater-than** | **less-than** | **not-equal**)—Match an operand.
- **value** ***sequence-number***—Match a decimal value.

  Range: 0 through 65,535

Required Privilege Level

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127

type (Security IDP ICMPv6 Headers)

Syntax

```
type {
  match (equal | greater-than | less-than | not-equal);
  value type-value;
}
```

Hierarchy Level

```
[edit security idp custom-attack attack-name attack-type signature protocol icmpv6]
```

Release Information

Statement introduced in Junos OS Release 12.3X48-D30.

Description

Specify the primary code that identifies the function of the request/reply.

Options

match (**equal** | **greater-than** | **less-than** | **not-equal**)—Match an operand.

value ***type-value***—Match a decimal value.

Range: 0 through 255

Required Privilege Level

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

Related Documentation

- IDP Extended Package Configuration Overview on page 127