Platform Settings

Setting the Date and Time

Use the NTP page to configure the network time protocol (NTP) on the system, to set the date and time manually, or to view the current system time.

NTP settings are automatically synced between the Firepower 4100/9300 chassis and any logical devices installed on the chassis.

Note If you are deploying threat defense on the Firepower 4100/9300 chassis, you must configure NTP on the Firepower 4100/9300 chassis so that Smart Licensing will work properly and to ensure proper timestamps on device registrations. You should use the same NTP server for both the Firepower 4100/9300 chassis and the management center, but note that you cannot use management center as the NTP server for the Firepower 4100/9300 chassis.

If you are using NTP, you can view the overall synchronization status on the **Current Time** tab, or you can view the synchronization status for each configured NTP server by looking at the Server Status field in the

NTP Server table on the **Time Synchronization** tab. If the system is unable to synchronize with a particular NTP server, you can hover over the information icon next to the Server Status for more information.

Viewing the Configured Date and Time

Procedure

Step 1 Choose **Platform Settings** > **NTP**.

Step 2 Click the **Current Time** tab.

The system shows the date, time, and time zone that are configured on the device.

If you are using NTP, you can also view the overall synchronization status on the **Current Time** tab. You can view the synchronization status for each configured NTP server by looking at the Server Status field in the **NTP Server** table on the **Time Synchronization** tab. If the system is unable to synchronize with a particular NTP server, you can hover over the information icon next to the Server Status for more information.

Setting the Time Zone

Procedure

Step 1 Choose **Platform Settings** > **NTP**.

Step 2 Click the **Current Time** tab.

Step 3 Choose the appropriate time zone for the chassis from the **Time Zone** drop-down list.

Setting the Date and Time Using NTP

NTP is used to implement a hierarchical system of servers that provide a precisely synchronized time among network systems. This kind of accuracy is required for time-sensitive operations, such as validating CRLs, which include a precise time stamp. You can configure up to four NTP servers.

Note

- FXOS uses NTP version 3.
- If the stratum value of an external NTP server is 13 or greater, FXOS rejects the NTP server and the server will be marked as failed. Thus, synchronization between the application instance and the NTP server is not possible on the FXOS chassis.

  If you have set up your own NTP server, you can find its stratum value in the /etc/ntp.conf file on the server. If the NTP server has stratum value of 13 or greater you can either change the stratum value in the ntp.conf file and restart the server or use a different NTP server (for example: pool.ntp.org). Once the NTP server stratum value is configured less than 13, you must remove the NTP server configuration and add it back on FXOS chassis to resync the application instance with NTP server.

Before you begin

If you use a hostname for the NTP server, you must configure a DNS server. See Configuring DNS Servers, on page 53.

Procedure

Step 1 Choose **Platform Settings** > **NTP**.

The **Time Synchronization** tab is selected by default.

Step 2 Under **Set Time Source**, click **Use NTP Server**.

Step 3 (Optional) Check the **NTP Server Authentication: Enable** check box if you need to authenticate with the NTP server.

Click **Yes** to require an authentication key ID and value.

Only SHA1 is supported for NTP server authentication.

Step 4 Click **Add** to identify up to 4 NTP servers by IP address or hostname.

Step 5 (Optional) Enter the NTP server's **Authentication Key** ID and **Authentication Value**.

Obtain the key ID and value from the NTP server. For example, to generate the SHA1 key on NTP server Version 4.2.8p8 or later with OpenSSL installed, enter the **ntp-keygen -M** command, and then view the key ID and value in the ntp.keys file. The key is used to tell both the client and server which value to use when computing the message digest.

Step 6 Click **Save**.

You can view the synchronization status of each server by looking at the Server Status field in the **NTP Server** table. If the system is unable to synchronize with a particular NTP server, you can hover over the information icon next to the Server Status for more information.

Note If you modify the system time by more than 10 minutes, the system will log you out and you will need to log in to the chassis manager again.

Note NTP warning **unreachable or Invalid NTP server** appears when the server state is unreachable. You can wait between 10 and 15 mins before attempting another server connection.

Deleting an NTP Server

Procedure

Step 1 Choose **Platform Settings** > **NTP**.

Step 2 Click the **Time Synchronization** tab.

Step 3 For each NTP server that you want to remove, click the **Delete** icon for that server in the **NTP Server** table.

Step 4 Click **Save**.

Setting the Date and Time Manually

This section describes how to set the date and time manually on the chassis. Note that after you manually set the chassis date and time, it could take some time for the change to be reflected in the installed logical device(s).

Procedure

Step 1 Choose **Platform Settings** > **NTP**.

Step 2 Click the **Time Synchronization** tab.

Step 3 Under **Set Time Source**, click **Set Time Manually**.

Step 4 Click the **Date** drop-down list to display a calendar and then set the date using the controls available in the calendar.

Step 5 Use the corresponding drop-down lists to specify the time as hours, minutes, and AM/PM.

Tip You can click **Get System Time** to set the date and time to match what is configured on the system you are using to connect to the chassis manager.

Step 6 Click **Save**.

The chassis is configured with the date and time specified.

Note If you modify the system time by more than 10 minutes, the system will log you out and you will need to log in to the chassis manager again.

Configuring SSH

The following procedure describes how to enable or disable SSH access to the chassis, how to enable the FXOS chassis as an SSH client, and how to configure the various algorithms used by SSH for encryption, key exchange, and message authentication for both the SSH server and SSH client.

SSH is enabled by default.

Procedure

Step 1 Choose **Platform Settings** > **SSH** > **SSH Server**.

Step 2 To enable SSH access to the chassis, check the **Enable SSH** check box. To disable SSH access, uncheck the **Enable SSH** check box.

Step 3 For the server **Encryption Algorithm**, check the check boxes for each allowed encryption algorithm.

Note

- The following encryption algorithms are not supported in Common Criteria mode:
  - 3des-cbc
  - chacha20-poly1305@openssh.com
- chacha20-poly1305@openssh.com is not supported in FIPS. If FIPS mode is enabled on the FXOS chassis, you cannot use chacha20-poly1305@openssh.com as an encryption algorithm.
- The following encryption algorithms are not enabled by default:

```
aes128-cbc
aes192-cbc
aes265-cbc
```

Step 4 For the server **Key Exchange Algorithm**, check the check boxes for each allowed Diffie-Hellman (DH) key exchange. The DH key exchange provides a shared secret that cannot be determined by either party alone. The key exchange is combined with a signature and the host key to provide host authentication. This key-exchange method provides explicit server authentication. For more information about using DH key-exchange methods, see RFC 4253.

Note

- The following key exchange algorithms are not supported in Common Criteria mode:
  - diffie-hellman-group14-sha256
  - curve25519-sha256
  - curve25519-sha256@libssh.org
- The following key exchange algorithms are not supported in FIPS mode:
  - curve25519-sha256
  - curve25519-sha256@libssh.org

Step 5 For the server **Mac Algorithm**, check the check boxes for each allowed integrity algorithm.

Step 6 For the server **Host Key**, enter the modulus size for the RSA key pairs.

The modulus value (in bits) is in multiples of 8 from 1024 to 2048. The larger the key modulus size you specify, the longer it takes to generate an RSA key pair. We recommend a value of 2048.

Step 7 For the server **Volume Rekey Limit**, set the amount of traffic in KB allowed over the connection before FXOS disconnects from the session.

Step 8 For the server **Time Rekey Limit**, set the minutes for how long an SSH session can be idle before FXOS disconnects the session.

Step 9 Click **Save**.

Step 10 Click the **SSH Client** tab to customize the FXOS chassis SSH client.

Step 11 For the **Strict Host Keycheck**, choose **enable**, **disable**, or **prompt** to control SSH host key checking.

- **enable**-The connection is rejected if the host key is not already in the FXOS known hosts file. You must manually add hosts at the FXOS CLI using the **enter ssh-host** command in the system/services scope.
- **prompt**-You are prompted to accept or reject the host key if it is not already stored on the chassis.

- **disable**-(The default) The chassis accepts the host key automatically if it was not stored before.

Step 12 For the client **Encryption Algorithm**, check the check boxes for each allowed encryption algorithm.

Note
- The following encryption algorithms are not supported in Common Criteria mode:
  - 3des-cbc
  - chacha20-poly1305@openssh.com

  If Common Criteria mode is enabled on the FXOS chassis, you cannot use 3des-cbc as an encryption algorithm.
- chacha20-poly1305@openssh.com is not supported in FIPS. If FIPS mode is enabled on the FXOS chassis, you cannot use chacha20-poly1305@openssh.com as an encryption algorithm.
- The following encryption algorithms are not enabled by default:

```
aes128-cbc
aes192-cbc
aes265-cbc
```

Step 13 For the client **Key Exchange Algorithm**, check the check boxes for each allowed Diffie-Hellman (DH) key exchange. The DH key exchange provides a shared secret that cannot be determined by either party alone. The key exchange is combined with a signature and the host key to provide host authentication. This key-exchange method provides explicit server authentication. For more information about using DH key-exchange methods, see RFC 4253.

Note
- The following Key Exchange Algorithms are not supported in Common Criteria mode:
  - diffie-hellman-group14-sha256
  - curve25519-sha256
  - curve25519-sha256@libssh.org
- The following Key Exchange Algorithms are not supported in FIPS mode:
  - curve25519-sha256
  - curve25519-sha256@libssh.org

Step 14 For the client **Mac Algorithm**, check the check boxes for each allowed integrity algorithm.

Step 15 For the client **Volume Rekey Limit**, set the amount of traffic in KB allowed over the connection before FXOS disconnects from the session.

Step 16 For the client **Time Rekey Limit**, set the minutes for how long an SSH session can be idle before FXOS disconnects the session.

Step 17 Click **Save**.

Configuring TLS

The Transport Layer Security (TLS) protocol provides privacy and data integrity between two communicating applications. You can use the FXOS CLI to configure the minimum TLS version allowed when the FXOS chassis communicates with external devices. Newer TLS versions provide more secure communications, older TLS versions allow for backward compatibility with older applications.

For example, if the minimum TLS version configured on your FXOS chassis is v1.1, and a client browser is configured to only run v1.0, then the client will not be able to open a connection with the FXOS Chassis Manager via HTTPS. As such, peer applications and LDAP servers must be configured appropriately.

This procedure shows how to configure and view the minimum version of TLS allowed for communication between FXOS chassis and an external device.

Note

- As of the FXOS 2.3(1) release, the default minimum TLS version for the FXOS chassis is v1.1.

Procedure

Step 1 Enter system mode:

Firepower-chassis# **scope system**

Step 2 View the TLS version options available to your system:

Firepower-chassis /system # **set services tls-ver**

Example:

```
Firepower-chassis /system #
Firepower-chassis /system # set services tls-ver
    v1_0  v1.0
    v1_1  v1.1
    v1_2  v1.2
```

Step 3 Set the minimum TLS version:

Firepower-chassis /system # **set services tls-ver** *version*

Example:

```
Firepower-chassis /system #
Firepower-chassis /system # set services tls-ver v1_2
```

Step 4 Commit the configuration:

Firepower-chassis /system # **commit-buffer**

Step 5 Show the minimum TLS version configured on your system:

Firepower-chassis /system # **scope services**

Firepower-chassis /system/services # **show**

Example:

```
Firepower-chassis /system/services # show
Name: ssh
    Admin State: Enabled
    Port: 22
Kex Algo: Diffie Hellman Group1 Sha1,Diffie Hellman Group14 Sha1
Mac Algo: Hmac Sha1,Hmac Sha1 96,Hmac Sha2 512,Hmac Sha2 256
Encrypt Algo: 3des Cbc,Aes256 Cbc,Aes128 Cbc,Aes192 Cbc,Aes256 Ctr,Aes128 Ctr,Ae
s192 Ctr
Auth Algo: Rsa
    Host Key Size: 2048
Volume: None Time: None
Name: telnet
    Admin State: Disabled
    Port: 23
Name: https
    Admin State: Enabled
    Port: 443
    Operational port: 443
    Key Ring: default
    Cipher suite mode: Medium Strength
    Cipher suite: ALL:!ADH:!EXPORT40:!EXPORT56:!LOW:!RC4:!MD5:!IDEA:+HIGH:+MEDIU
M:+EXP:+eNULL
    Https authentication type: Cert Auth
    Crl mode: Relaxed
TLS:
    TLS version: v1.2
```

Configuring Telnet

The following procedure describes how to enable or disable Telnet access to the chassis. Telnet is disabled by default.

Note Telnet configuration is currently only available using the CLI.

Procedure

Step 1 Enter system mode:

Firepower-chassis # **scope system**

Step 2 Enter system services mode:

Firepower-chassis /system # **scope services**

Step 3 To configure Telnet access to the chassis, do one of the following:

- To allow Telnet access to the chassis, enter the following command:

  Firepower-chassis /system/services # **enable telnet-server**

- To disallow Telnet access to the chassis, enter the following command:

  Firepower-chassis /system/services # **disable telnet-server**

Step 4 Commit the transaction to the system configuration:

Firepower /system/services # **commit-buffer**

Example

The following example enables Telnet and commits the transaction:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /services # enable telnet-server
Firepower-chassis /services* # commit-buffer
Firepower-chassis /services #
```

Configuring SNMP

Use the SNMP page to configure the Simple Network Management Protocol (SNMP) on the chassis. See the following topics for more information:

About SNMP

The Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

The SNMP framework consists of three parts:

- An SNMP manager—The system used to control and monitor the activities of network devices using SNMP.
- An SNMP agent—The software component within the chassis that maintains the data for the chassis and reports the data, as needed, to the SNMP manager. The chassis includes the agent and a collection of MIBs. To enable the SNMP agent and create the relationship between the manager and agent, enable and configure SNMP in the chassis manager or the FXOS CLI.
- A managed information base (MIB)—The collection of managed objects on the SNMP agent.

The chassis supports SNMPv1, SNMPv2c and SNMPv3. Both SNMPv1 and SNMPv2c use a community-based form of security. SNMP is defined in the following:

- RFC 3410 (http://tools.ietf.org/html/rfc3410)
- RFC 3411 (http://tools.ietf.org/html/rfc3411)
- RFC 3412 (http://tools.ietf.org/html/rfc3412)
- RFC 3413 (http://tools.ietf.org/html/rfc3413)
- RFC 3414 (http://tools.ietf.org/html/rfc3414)
- RFC 3415 (http://tools.ietf.org/html/rfc3415)
- RFC 3416 (http://tools.ietf.org/html/rfc3416)

- RFC 3417 (http://tools.ietf.org/html/rfc3417)
- RFC 3418 (http://tools.ietf.org/html/rfc3418)
- RFC 3584 (http://tools.ietf.org/html/rfc3584)

Note Be aware that SNMP versions 1 and 2c have serious known security issues: they transmit all information without encryption, including the community string, which serves as the only form of authentication in these versions.

SNMP Notifications

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

The chassis generates SNMP notifications as either traps or informs. Traps are less reliable than informs because the SNMP manager does not send any acknowledgment when it receives a trap, and the chassis cannot determine if the trap was received. An SNMP manager that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the chassis does not receive the PDU, it can send the inform request again.

However, informs are available only with SNMPv2c, which is considered insecure, and is not recommended.

Note The ifindex order on the interface that uses SNMP does not change after you reboot the FXOS. However, the index number on the FXOS disk usage OID changes when you reboot the FXOS.

SNMP Security Levels and Privileges

SNMPv1, SNMPv2c, and SNMPv3 each represent a different security model. The security model combines with the selected security level to determine the security mechanism applied when the SNMP message is processed.

The security level determines the privileges required to view the message associated with an SNMP trap. The privilege level determines whether the message needs to be protected from disclosure or authenticated. The supported security level depends upon which security model is implemented. SNMP security levels support one or more of the following privileges:

- noAuthNoPriv—No authentication or encryption
- authNoPriv—Authentication but no encryption
- authPriv—Authentication and encryption

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy that is set up for a user and the role in which the user resides. A security level is the permitted level of security

within a security model. A combination of a security model and a security level determines which security mechanism is employed when handling an SNMP packet.

Supported Combinations of SNMP Security Models and Levels

The following table identifies what the combinations of security models and levels mean.

Table 1: SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication. **Note** While you can configure it, FXOS does not support use of `noAuthNoPriv` with SNMP version 3.
v3	authNoPriv	HMAC-SHA	No	Provides authentication based on the HMAC Secure Hash Algorithm (SHA).
v3	authPriv	HMAC-SHA	DES	Provides authentication based on the HMAC-SHA algorithm. Provides Data Encryption Standard (DES) 56-bit encryption in addition to authentication based on the Cipher Block Chaining (CBC) DES (DES-56) standard.

SNMPv3 Security Features

SNMPv3 provides secure access to devices by a combination of authenticating and encrypting frames over the network. SNMPv3 authorizes management operations only by configured users and encrypts SNMP messages. The SNMPv3 User-Based Security Model (USM) refers to SNMP message-level security and offers the following services:

- Message integrity—Ensures that messages have not been altered or destroyed in an unauthorized manner and that data sequences have not been altered to an extent greater than can occur non-maliciously.
- Message origin authentication—Ensures that the claimed identity of the user on whose behalf received data was originated is confirmed.
- Message confidentiality and encryption—Ensures that information is not made available or disclosed to unauthorized individuals, entities, or processes.

SNMP Support

The chassis provides the following support for SNMP:

Support for MIBs

The chassis supports read-only access to MIBs.

For information about the specific MIBs available and where you can obtain them, see the Cisco FXOS MIB Reference Guide.

Authentication Protocol for SNMPv3 Users

The chassis supports the HMAC-SHA-96 (SHA) authentication protocol for SNMPv3 users.

AES Privacy Protocol for SNMPv3 Users

The chassis uses Advanced Encryption Standard (AES) as one of the privacy protocols for SNMPv3 message encryption and conforms with RFC 3826.

The privacy password, or priv option, offers a choice of DES or 128-bit AES encryption for SNMP security encryption. If you enable AES-128 configuration and include a privacy password for an SNMPv3 user, the chassis uses the privacy password to generate a 128-bit AES key. The AES privacy password can have a minimum of eight characters. If the passphrases are specified in clear text, you can specify a maximum of 64 characters.

Enabling SNMP and Configuring SNMP Properties

Procedure

Step 1 Choose **Platform Settings** > **SNMP**.

Step 2 In the **SNMP** area, complete the following fields:

Name	Description
Admin Instance drop-down menu	If MIO is the manager, **Native** is displayed by default. To configure SNMP unification, from the drop-down, select the threat defense instance or ASA, if any. **Note** When you select a threat defense instance or ASA other than **Native** (MIO) as the manager, all fields on this page are dimmed. **Important** After configuring SNMP unification, wait for 5 minutes before you proceed with SNMP polling.
Admin State check box	Whether SNMP is enabled or disabled. Enable this service only if your system includes integration with an SNMP server.
Port field	The port on which the chassis communicates with the SNMP host. You cannot change the default port.

Name	Description
Community/Username field	(Optional) The community string used for polling in SNMP v1 and v2. When you specify an SNMP community name, you are also automatically enabling SNMP versions 1 and 2c for polling requests from the SNMP remote manager. This field is not applicable to SNMP v3. Be aware that SNMP versions 1 and 2c have serious known security issues: they transmit all information without encryption, including the community string, which serves as the only form of authentication in these versions. Enter an alphanumeric string between 1 and 32 characters. Do not use @ (at sign), & (ampersand), ? (question mark) or an empty space. The default is **`public`**. If the **Community/Username** field is already set, the text to the right of the empty field reads **Set: Yes**. If the **Community/Username** field is not yet populated with a value, the text to the right of the empty field reads **Set: No**. **Note** You can use the CLI command **set snmp community** to delete an existing community string, thereby disabling SNMP versions 1 and 2c for polling requests from the SNMP remote manager.
System Administrator Name field	The contact person responsible for the SNMP implementation. Enter a string of up to 255 characters, such as an email address or a name and telephone number.
Location field	The location of the host on which the SNMP agent (server) runs. Enter an alphanumeric string up to 510 characters.

Step 3 Click **Save**.

What to do next

Create SNMP traps and users.

Creating an SNMP Trap

The following procedure describes how to create SNMP traps.

Note You can define up to eight SNMP traps.

Procedure

Step 1 Choose **Platform Settings** > **SNMP**.

Step 2 In the **SNMP Traps** area, click **Add**.

Step 3 In the **Add SNMP Trap** dialog box, complete the following fields:

Name	Description
Host Name field	The hostname or IP address of the SNMP host to which the chassis should send the traps.
Community/Username field	Enter the SNMPv1/v2c community string, or the SNMPv3 user name, needed to permit access to the trap destination. This must be the same as the community or user name that is configured for the SNMP service. Enter an alphanumeric string between 1 and 32 characters. Do not use @ (at sign), \ (backslash), " (double quote), ? (question mark) or an empty space.
Port field	The port on which the chassis communicates with the SNMP host for the trap. Enter an integer between 1 and 65535.
Version field	The SNMP version and model used for the trap. This can be one of the following: • **V1** • **V2** • **V3** **Note** Be aware that SNMP versions 1 and 2c have serious known security issues: they transmit all information without encryption, including the community string, which serves as the only form of authentication in these versions.
Type field	Specify the type of trap to send: • **Traps** • **Informs** (only valid when **Version** is **V2**)
v3 Privilege field	If you selected **V3** for the version, specify the privilege level associated with the trap: • **Auth**—Authentication but no encryption. • **Noauth**—No authentication or encryption. Note that while you can select it, FXOS does not support this security level with SNMPv3. • **Priv**—Authentication and encryption.

Step 4 Click **OK** to close the **Add SNMP Trap** dialog box.

Step 5 Click **Save**.

Deleting an SNMP Trap

Procedure

Step 1 Choose **Platform Settings** > **SNMP**.

Step 2 In the **SNMP Traps** area, click the **Delete** icon in the row in the table that corresponds to the trap you want to delete.

Creating an SNMPv3 User

Procedure

Step 1 Choose **Platform Settings** > **SNMP**.

Step 2 In the **SNMP Users** area, click **Add**.

Step 3 In the **Add SNMP User** dialog box, complete the following fields:

Name	Description
Name field	The user name assigned to the SNMPv3 user. Enter up to 32 characters. The name must begin with a letter. Valid characters include letters, numbers, _ (underscore), . (period), @ (at sign), and - (hyphen).
Auth Type field	The authorization type: **SHA**.
Use AES-128 check box	If checked, this user uses AES-128 encryption. **Note** SNMPv3 does not support DES. If you leave the AES-128 box unchecked, no privacy encryption will be done and any configured privacy password will have no effect.

Name	Description
Password field	The password for this user. The FXOS rejects any password that does not meet the following requirements: • Must contain a minimum of 8 characters and a maximum of 80 characters. • Must contain only letters, numbers, and the following characters: ~`!@#%^&*()_-+{}[]|\:;"'<,>./ • Must not contain the following symbols: $ (dollar sign), ? (question mark), or = (equals sign). • Must contain at least five different characters. • Must not contain too many consecutively incrementing or decrementing numbers or letters. For example, the string "12345" has four such characters, and the string "ZYXW" has three. If the total number of such characters exceeds a certain limit (typically more than around 4-6 such occurrences), the simplicity check will fail. **Note** The consecutively incrementing or decrementing character count is not reset when non-incrementing or decrementing characters are used in between. For example, abcd&!21 will fail the password check, but abcd&!25, will not.
Confirm Password field	The password again for confirmation purposes.

Name	Description
Privacy Password field	The privacy password for this user. The FXOS rejects any password that does not meet the following requirements: • Must contain a minimum of 8 characters and a maximum of 80 characters. • Must contain only letters, numbers, and the following characters: ~`!@#%^&*()_-+{}[]|\:;"'<,>./ • Must not contain the following symbols: $ (dollar sign), ? (question mark), or = (equals sign). • Must contain at least five different characters. • Must not contain too many consecutively incrementing or decrementing numbers or letters. For example, the string "12345" has four such characters, and the string "ZYXW" has three. If the total number of such characters exceeds a certain limit (typically more than around 4-6 such occurrences), the simplicity check will fail. **Note** The consecutively incrementing or decrementing character count is not reset when non-incrementing or decrementing characters are used in between. For example, abcd&!21 will fail the password check, but abcd&!25, will not.
Confirm Privacy Password field	The privacy password again for confirmation purposes.

Step 4 Click **OK** to close the **Add SNMP User** dialog box.

Step 5 Click **Save**.

Deleting an SNMPv3 User

Procedure

Step 1 Choose **Platform Settings** > **SNMP**.

Step 2 In the **SNMP Users** area, click the **Delete** icon in the row in the table that corresponds to the user you want to delete.

Configuring HTTPS

This section describes how to configure HTTPS on the Firepower 4100/9300 chassis.

Note You can change the HTTPS port using chassis manager or the FXOS CLI. All other HTTPS configuration can only be done using the FXOS CLI.

Certificates, Key Rings, and Trusted Points

HTTPS uses components of the Public Key Infrastructure (PKI) to establish secure communications between two devices, such as a client's browser and the Firepower 4100/9300 chassis.

Encryption Keys and Key Rings

Each PKI device holds a pair of asymmetric Rivest-Shamir-Adleman (RSA) encryption keys, one kept private and one made public, stored in an internal key ring. A message encrypted with either key can be decrypted with the other key. To send an encrypted message, the sender encrypts the message with the receiver's public key, and the receiver decrypts the message using its own private key. A sender can also prove its ownership of a public key by encrypting (also called 'signing') a known message with its own private key. If a receiver can successfully decrypt the message using the public key in question, the sender's possession of the corresponding private key is proven. Encryption keys can vary in length, with typical lengths from 512 bits to 2048 bits. In general, a longer key is more secure than a shorter key. FXOS provides a default key ring with an initial 2048-bit key pair, and allows you to create additional key rings.

The default key ring certificate must be manually regenerated if the cluster name changes or the certificate expires.

Certificates

To prepare for secure communications, two devices first exchange their digital certificates. A certificate is a file containing a device's public key along with signed information about the device's identity. To merely support encrypted communications, a device can generate its own key pair and its own self-signed certificate. When a remote user connects to a device that presents a self-signed certificate, the user has no easy method to verify the identity of the device, and the user's browser will initially display an authentication warning. By default, FXOS contains a built-in self-signed certificate containing the public key from the default key ring.

Trusted Points

To provide stronger authentication for FXOS, you can obtain and install a third-party certificate from a trusted source, or trusted point, that affirms the identity of your device. The third-party certificate is signed by the issuing trusted point, which can be a root certificate authority (CA) or an intermediate CA or trust anchor that is part of a trust chain that leads to a root CA. To obtain a new certificate, you must generate a certificate request through FXOS and submit the request to a trusted point.

Important The certificate must be in Base64 encoded X.509 (CER) format.

Creating a Key Ring

FXOS supports a maximum of 8 key rings, including the default key ring.

Procedure

Step 1 Enter security mode:

Firepower-chassis # **scope security**

Step 2 Create and name the key ring:

Firepower-chassis # **create keyring** *keyring-name*

Step 3 Set the SSL key length in bits:

Firepower-chassis # **set modulus** {**mod1024** | **mod1536** | **mod2048** | **mod512**}

Step 4 Commit the transaction:

Firepower-chassis # **commit-buffer**

Example

The following example creates a keyring with a key size of 1024 bits:

```
Firepower-chassis# scope security
Firepower-chassis /security # create keyring kr220
Firepower-chassis /security/keyring* # set modulus mod1024
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

What to do next

Create a certificate request for this key ring.

Regenerating the Default Key Ring

The default key ring certificate must be manually regenerated if the cluster name changes or the certificate expires.

Note The default keyring is only used by FCM on FXOS.

Procedure

Step 1 Enter security mode:

Firepower-chassis # **scope security**

Step 2 Enter key ring security mode for the default key ring:

Firepower-chassis /security # **scope keyring default**

Step 3 Regenerate the default key ring:

Firepower-chassis /security/keyring # **set regenerate yes**

Step 4 Commit the transaction:

Firepower-chassis # **commit-buffer**

Example

The following example regenerates the default key ring:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring default
Firepower-chassis /security/keyring* # set regenerate yes
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

Creating a Certificate Request for a Key Ring

Creating a Certificate Request for a Key Ring with Basic Options

Procedure

Step 1 Enter security mode:

Firepower-chassis # **scope security**

Step 2 Enter configuration mode for the key ring:

Firepower-chassis /security # **scope keyring** *keyring-name*

Step 3 Create a certificate request using the IPv4 or IPv6 address specified, or the name of the fabric interconnect. You are prompted to enter a password for the certificate request.

Firepower-chassis /security/keyring # **create certreq** {**ip** [*ipv4-addr* | *ipv6-v6*] |**subject-name** *name*}

Step 4 Commit the transaction:

Firepower-chassis /security/keyring/certreq # **commit-buffer**

Step 5 Display the certificate request, which you can copy and send to a trust anchor or certificate authority:

Firepower-chassis /security/keyring # **show certreq**

Example

The following example creates and displays a certificate request with an IPv4 address for a key ring, with basic options:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # create certreq ip 192.168.200.123 subject-name sjc04
Certificate request password:
Confirm certificate request password:
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request e-mail name:
Certificate request country name:
State, province or county (full name):
Locality (eg, city):
Organization name (eg, company):
Organization Unit name (eg, section):
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYW1jMDQwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKn1t8qMZO4UGqILKFXQQc2c8b/vW2rnRF8OPhKbhghLA1YZ1F
JqcYEG5Yl1+vgohLBTd45s0GC8m4RTLJWHo4SwccAUXQ5Zngf45YtX1WsylwUWV4
0re/zgTk/WCd56RfOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQQMA6CBnNhbWMwNIcECsEiXjAN
BgkqhkiG9w0BAQQFAAOBgQCsxN0qUHYGFoQw56RwQueLTNPnrndqUwuZHUO03Teg
nhsyu4satpyiPqVV9viKZ+spvc6x5PWIcTWgHhH8BimOb/0OKuG8kwfIGGsEDlAv
TTYvUP+BZ9OFiPbRIA718S+V8ndXr1HejiQGxlDNqoN+odCXPc5kjoXD0lZTL09H
BA==
-----END CERTIFICATE REQUEST-----

Firepower-chassis /security/keyring #
```

What to do next

- Copy the text of the certificate request, including the BEGIN and END lines, and save it in a file. Send the file with the certificate request to a trust anchor or certificate authority to obtain a certificate for the key ring.
- Create a trusted point and set the certificate chain for the certificate of trust received from the trust anchor.

Creating a Certificate Request for a Key Ring with Advanced Options

Procedure

Step 1 Enter security mode:

Firepower-chassis # **scope security**

Step 2 Enter configuration mode for the key ring:

Firepower-chassis /security # **scope keyring** *keyring-name*

Step 3 Create a certificate request:

Firepower-chassis /security/keyring # **create certreq**

Step 4 Specify the country code of the country in which the company resides:

Firepower-chassis /security/keyring/certreq* # **set country** *country name*

Step 5 Specify the Domain Name Server (DNS) address associated with the request:

Firepower-chassis /security/keyring/certreq* # **set dns** *DNS Name*

Step 6 Specify the email address associated with the certificate request:

Firepower-chassis /security/keyring/certreq* # **set e-mail** *E-mail name*

Step 7 Specify the IP address of the Firepower 4100/9300 chassis:

Firepower-chassis /security/keyring/certreq* # **set ip** {*certificate request ip-address*|*certificate request ip6-address* }

Step 8 Specify the city or town in which the company requesting the certificate is headquartered:

Firepower-chassis /security/keyring/certreq* # **set locality** *locality name (eg, city)*

Step 9 Specify the organization requesting the certificate:

Firepower-chassis /security/keyring/certreq* # **set org-name** *organization name*

Step 10 Specify the organizational unit:

Firepower-chassis /security/keyring/certreq* # **set org-unit-name** *organizational unit name*

Step 11 Specify an optional password for the certificate request:

Firepower-chassis /security/keyring/certreq* # **set password** *certificate request password*

Step 12 Specify the state or province in which the company requesting the certificate is headquartered:

Firepower-chassis /security/keyring/certreq* # **set state** *state, province or county*

Step 13 Specify the fully qualified domain name of the Firepower 4100/9300 chassis:

Firepower-chassis /security/keyring/certreq* # **set subject-name** *certificate request name*

Step 14 Commit the transaction:

Firepower-chassis /security/keyring/certreq # **commit-buffer**

Step 15 Display the certificate request, which you can copy and send to a trust anchor or certificate authority:

Firepower-chassis /security/keyring # **show certreq**

Example

Note We recommend not to commit buffer with a "set dns" or "set subject-name" without FQDN for releases earlier than 2.7. If you try to create a certification requirement with a DNS or subject name that is not a FQDN, it will throw an error.

The following example creates and displays a certificate request with an IPv4 address for a key ring, with advanced options:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # create certreq
Firepower-chassis /security/keyring/certreq* # set "ip 192.168.200.123"
Firepower-chassis /security/keyring/certreq* #  set subject-name "sjc04"
Firepower-chassis /security/keyring/certreq* # set country "US"
Firepower-chassis /security/keyring/certreq* # set dns "bg1-samc-15A"
Firepower-chassis /security/keyring/certreq* # set email "test@cisco.com"
Firepower-chassis /security/keyring/certreq* # set locality "new york city"
Firepower-chassis /security/keyring/certreq* # set org-name "Cisco Systems"
Firepower-chassis /security/keyring/certreq* # set org-unit-name "Testing"
Firepower-chassis /security/keyring/certreq* # set state "new york"
Firepower-chassis /security/keyring/certreq* # commit-buffer
Firepower-chassis /security/keyring/certreq # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request e-mail name: test@cisco.com
Certificate request country name: US
State, province or county (full name): New York
Locality name (eg, city): new york city
Organization name (eg, company): Cisco
Organization Unit name (eg, section): Testing
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYW1jMDQwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKn1t8qMZO4UGqILKFXQQc2c8b/vW2rnRF8OPhKbhghLA1YZ1F
JqcYEG5Yl1+vgohLBTd45s0GC8m4RTLJWHo4SwccAUXQ5Zngf45YtX1WsylwUWV4
0re/zgTk/WCd56RfOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQQMA6CBnNhbWMwNIcECsEiXjAN
BgkqhkiG9w0BAQQFAAOBgQCsxN0qUHYGFoQw56RwQueLTNPnrndqUwuZHUO03Teg
nhsyu4satpyiPqVV9viKZ+spvc6x5PWIcTWgHhH8BimOb/0OKuG8kwfIGGsEDlAv
TTYvUP+BZ9OFiPbRIA718S+V8ndXr1HejiQGxlDNqoN+odCXPc5kjoXD0lZTL09H
BA==
-----END CERTIFICATE REQUEST-----

Firepower-chassis /security/keyring/certreq #
```

What to do next

- Copy the text of the certificate request, including the BEGIN and END lines, and save it in a file. Send the file with the certificate request to a trust anchor or certificate authority to obtain a certificate for the key ring.
- Create a trusted point and set the certificate chain for the certificate of trust received from the trust anchor.

Creating a Trusted Point

Procedure

Step 1 Enter security mode:

Firepower-chassis # **scope security**

Step 2 Create a trusted point:

Firepower-chassis /security # **create trustpoint** *name*

Step 3 Specify certificate information for this trusted point:

Firepower-chassis /security/trustpoint # **set certchain** [*certchain*]

If you do not specify certificate information in the command, you are prompted to enter a certificate or a list of trustpoints defining a certification path to the root certificate authority (CA). On the next line following your input, type **ENDOFBUF** to finish.

Important The certificate must be in Base64 encoded X.509 (CER) format.

Step 4 Commit the transaction:

Firepower-chassis /security/trustpoint # **commit-buffer**

Example

The following example creates a trusted point and provides a certificate for the trusted point:

```
Firepower-chassis# scope security
Firepower-chassis /security # create trustpoint tPoint10
Firepower-chassis /security/trustpoint* # set certchain
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Trustpoint Certificate Chain:
> -----BEGIN CERTIFICATE-----
> MIIDMDCCApmgAwIBAgIBADANBgkqhkiG9w0BAQQFADB0MQswCQYDVQQGEwJVUzEL
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGxlIEluYy4xEzARBgNVBAsT
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWEHVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMivyCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBNKONDl
> GMbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bF5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzCl903O6Mg51zq1zXcz75+VFj2I6rH9asckCld3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtvlWvfhevskV0j6
> jtcEMyZ+f7+3yh421ido3nO4MIGeBgNVHSMEgZYwgZOAFLlNjtcEMyZ+f7+3yh42
> 1ido3nO4oXikdjB0MQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExFDASBgNVBAcT
> C1NhbnRhIENsYXJhMRswGQYDVQQKExJOdW92YSBTeXN0ZW1zIEluYy4xFDASBgNV
> BAsTC0VuZ2luZWVyaW5nMQ8wDQYDVQQDEwZ0ZXN0Q0GCAQAwDAYDVR0TBAUwAwEB
> /zANBgkqhkiG9w0BAQQFAAOBgQAhWaRwXNR6B4g6Lsnr+fptHv+WVhB5fKqGQqXc
> wR4pYiO4z42/j9Ijenh75tCKMhW51az8copP1EBmOcyuhf5C6vasrenn1ddkkYt4
> PR0vxGc40whuiozBolesmsmjBbedUCwQgdFDWhDIZJwK5+N3x/kfa2EHU6id1avt
> 4YL5Jg==
> -----END CERTIFICATE-----
```

```
> ENDOFBUF
Firepower-chassis /security/trustpoint* # commit-buffer
Firepower-chassis /security/trustpoint #
```

What to do next

Obtain a key ring certificate from the trust anchor or certificate authority and import it into the key ring.

Importing a Certificate into a Key Ring

Before you begin

- Configure a trusted point that contains the certificate chain for the key ring certificate.
- Obtain a key ring certificate from a trust anchor or certificate authority.

Note If you change the certificate in a key ring that has already been configured on HTTPS, you must restart HTTPS in order for the new certificate to take effect. For more informaiton, see: Restarting HTTPS, on page 28.

Procedure

Step 1 Enter security mode:

Firepower-chassis # **scope security**

Step 2 Enter configuration mode for the key ring that will receive the certificate:

Firepower-chassis /security # **scope keyring** *keyring-name*

Step 3 Specify the trusted point for the trust anchor or certificate authority from which the key ring certificate was obtained:

Firepower-chassis /security/keyring # **set trustpoint** *name*

Step 4 Launch a dialog for entering and uploading the key ring certificate:

Firepower-chassis /security/keyring # **set cert**

At the prompt, paste the certificate text that you received from the trust anchor or certificate authority. On the next line following the certificate, type **ENDOFBUF** to complete the certificate input.

Important The certificate must be in Base64 encoded X.509 (CER) format.

Step 5 Commit the transaction:

Firepower-chassis /security/keyring # **commit-buffer**

Example

The following example specifies the trust point and imports a certificate into a key ring:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope keyring kr220
Firepower-chassis /security/keyring # set trustpoint tPoint10
Firepower-chassis /security/keyring* # set cert
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Keyring certificate:
> -----BEGIN CERTIFICATE-----
> MIIB/zCCAWgCAQAwgZkxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJDQTEVMBMGA1UE
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGxlIEluYy4xEzARBgNVBAsT
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWEHVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMivyCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBNKONDl
> GMbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bF5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzCl903O6Mg51zq1zXcz75+VFj2I6rH9asckCld3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtvlWvfhevskV0j6
> mK3Ku+YiORnv6DhxrOoqau8r/hyI/L43l7IPN1HhOi3oha4=
> -----END CERTIFICATE-----
> ENDOFBUF
Firepower-chassis /security/keyring* # commit-buffer
Firepower-chassis /security/keyring #
```

What to do next

Configure your HTTPS service with the key ring.

Configuring HTTPS

Caution After you complete the HTTPS configuration, including changing the port and key ring to be used by HTTPS, all current HTTP and HTTPS sessions are closed without warning as soon as you save or commit the transaction.

Procedure

Step 1 Enter system mode:

Firepower-chassis# **scope system**

Step 2 Enter system services mode:

Firepower-chassis /system # **scope services**

Step 3 Enable the HTTPS service:

Firepower-chassis /system/services # **enable https**

Step 4 (Optional) Specify the port to be used for the HTTPS connection:

Firepower-chassis /system/services # **set https port** *port-num*

Step 5 (Optional) Specify the name of the key ring you created for HTTPS:

Firepower-chassis /system/services # **set https keyring** *keyring-name*

Step 6 (Optional) Specify the level of Cipher Suite security used by the domain:

Firepower-chassis /system/services # **set https cipher-suite-mode** *cipher-suite-mode*

cipher-suite-mode can be one of the following keywords:

- **high-strength**
- **medium-strength**
- **low-strength**
- **custom**—Allows you to specify a user-defined Cipher Suite specification string.

Step 7 (Optional) If **cipher-suite-mode** is set to **custom**, specify a custom level of Cipher Suite security for the domain:

Firepower-chassis /system/services # **set https cipher-suite** *cipher-suite-spec-string*

cipher-suite-spec-string can contain up to 256 characters and must conform to the OpenSSL Cipher Suite specifications. You cannot use any spaces or special characters except ! (exclamation point), + (plus sign), - (hyphen), and : (colon). For details, see http://httpd.apache.org/docs/2.0/mod/mod_ssl.html#sslciphersuite.

For example, the medium strength specification string FXOS uses as the default is:
`ALL:!ADH:!EXPORT56:!LOW:RC4+RSA:+HIGH:+MEDIUM:+EXP:+eNULL`

Note This option is ignored if **cipher-suite-mode** is set to anything other than **custom**.

Step 8 Commit the transaction to the system configuration:

Firepower-chassis /system/services # **commit-buffer**

Example

The following example enables HTTPS, sets the port number to 443, sets the key ring name to kring7984, sets the Cipher Suite security level to high, and commits the transaction:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # enable https
Firepower-chassis /system/services* # set https port 443
Warning: When committed, this closes all the web sessions.
Firepower-chassis /system/services* # set https keyring kring7984
Firepower-chassis /system/services* # set https cipher-suite-mode high
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

Changing the HTTPS Port

The HTTPS service is enabled on port 443 by default. You cannot disable HTTPS, but you can change the port to use for HTTPS connections.

Procedure

Step 1 Choose **Platform Settings** > **HTTPS**.

Step 2 Enter the port to use for HTTPS connections in the **Port** field. Specify an integer between 1 and 65535. This service is enabled on port 443 by default.

Step 3 Click **Save**.

The chassis is configured with the HTTPS port specified.

After changing the HTTPS port, all current HTTPS sessions are closed. Users will need to log back in to the chassis manager using the new port as follows:

`https://<chassis_mgmt_ip_address>:<chassis_mgmt_port>`

where <*chassis_mgmt_ip_address*> is the IP address or host name of the chassis that you entered during initial configuration and <*chassis_mgmt_port*> is the HTTPS port you have just configured.

Restarting HTTPS

If you change the certificate in a key ring that has already been configured on HTTPS, you must restart HTTPS in order for the new certificate to take effect. Use the following procedure to reset HTTPS with an updated keyring.

Procedure

Step 1 Enter system mode:

Firepower-chassis# **scope system**

Step 2 Enter system services mode:

Firepower-chassis /system # **scope services**

Step 3 Set the HTTPS key ring back to its default value:

Firepower-chassis /system/services # **set https keyring default**

Step 4 Commit the transaction to the system configuration:

Firepower-chassis /system/services # **commit-buffer**

Step 5 Wait five seconds.

Step 6 Set HTTPS with the key ring you created:

Firepower-chassis /system/services # **set https keyring** *keyring-name*

Step 7 Commit the transaction to the system configuration:

Firepower-chassis /system/services # **commit-buffer**

Deleting a Key Ring

Procedure

Step 1 Enter security mode:

Firepower-chassis # **scope security**

Step 2 Delete the named key ring:

Firepower-chassis /security # **delete keyring** *name*

Step 3 Commits the transaction:

Firepower-chassis /security # **commit-buffer**

Example

The following example deletes a key ring:

```
Firepower-chassis# scope security
Firepower-chassis /security # delete keyring key10
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

Deleting a Trusted Point

Before you begin

Ensure that the trusted point is not used by a key ring.

Procedure

Step 1 Enters security mode:

Firepower-chassis# **scope security**

Step 2 Delete the named trusted point:

Firepower-chassis /security # **delete trustpoint** *name*

Step 3 Commits the transaction:

Firepower-chassis /security # **commit-buffer**

Example

The following example deletes a trusted point:

```
Firepower-chassis# scope security
Firepower-chassis /security # delete trustpoint tPoint10
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

Disabling HTTPS

Procedure

Step 1 Enter system mode:

Firepower-chassis# **scope system**

Step 2 Enter system services mode:

Firepower-chassis /system # **scope services**

Step 3 Disable the HTTPS service:

Firepower-chassis /system/services # **disable https**

Step 4 Commit the transaction to the system configuration:

Firepower-chassis /system/services # **commit-buffer**

Example

The following example disables HTTPS and commits the transaction:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # disable https
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

Configuring AAA

This section describes authentication, authorization, and accounting. See the following topics for more information:

About AAA

Authentication, Authorization and Accounting (AAA) is a set of services for controlling access to network resources, enforcing policies, assessing usage, and providing the information necessary to bill for services.

Authentication identifies the user. Authorization implements policies that determine which resources and services an authenticated user may access. Accounting keeps track of time and data resources that are used for billing and analysis. These processes are considered important for effective network management and security.

Authentication

Authentication provides a way to identify each user, typically by having the user enter a valid user name and valid password before access is granted. The AAA server compares the user's provided credentials with user credentials stored in a database. If the credentials are matched, the user is permitted access to the network. If the credentials do not match, authentication fails and network access is denied.

You can configure the Firepower 4100/9300 chassis to authenticate administrative connections to the chassis, including the following sessions:

- HTTPS
- SSH
- Serial console

Authorization

Authorization is the process of enforcing policies: determining what types of activities, resources, or services each user is permitted to access. After authentication, a user may be authorized for different types of access or activity.

Accounting

Accounting measures the resources a user consumes during access, which may include the amount of system time or the amount of data that a user has sent or received during a session. Accounting is carried out through the logging of session statistics and usage information, which is used for authorization control, billing, trend analysis, resource utilization, and capacity planning activities.

Interaction Between Authentication, Authorization, and Accounting

You can use authentication alone, or with authorization and accounting. Authorization always requires a user to be authenticated first. You can use accounting alone, or with authentication and authorization.

Supported Types of Authentication

FXOS supports the following types of user Authentication:

- **Remote** – The following network AAA services are supported:
  - LDAP
  - RADIUS
  - TACACS+
  - Single Sign-On (SSO)
- **Local** – The chassis maintains a local database that you can populate with user profiles. You can use this local database instead of AAA servers to provide user authentication, authorization, and accounting.

User Roles

FXOS supports local and remote Authorization in the form of user-role assignment. The roles that can be assigned are:

- **Admin** – Complete read-and-write access to the entire system. The default admin account is assigned this role by default and it cannot be changed.
- **AAA Administrator** – Read-and-write access to users, roles, and AAA configuration. Read access to the rest of the system.
- **Operations** – Read-and-write access to NTP configuration, Smart Call Home configuration for Smart Licensing, and system logs, including syslog servers and faults. Read access to the rest of the system.
- **Read-Only** – Read-only access to system configuration with no privileges to modify the system state.

See User Management for more information about local users and role assignments.

Setting Up AAA

These steps provide a basic outline for setting up Authentication, Authorization and Accounting (AAA) on a Firepower 4100/9300 appliance.

1. Configure the desired type(s) of user authentication:
   - **Local** – User definitions and local authentication are part of User Management.
   - **Remote** – Configuring remote AAA server access is part of Platform Settings, specifically:
     - Configuring LDAP Providers, on page 33
     - Configuring RADIUS Providers, on page 36
     - Configuring TACACS+ Providers, on page 38
     - Configuring Single Sign-On (SSO), on page 40

Note If you will be using remote AAA servers, be sure to enable and configure AAA services on the remote servers before configuring remote AAA server access on the chassis.

2. Specify the default authentication method—this also is part of User Management.

Note If Default Authentication and Console Authentication are both set to use the same remote authentication protocol (RADIUS, TACACS+, or LDAP), you cannot change certain aspects of that server's configuration (for example, deleting that server, or changing its order of assignment) without updating these user settings.

Configuring LDAP Providers

Configuring Properties for LDAP Providers

The properties that you configure in this task are the default settings for all provider connections of this type. If an individual provider includes a setting for any of these properties, the FXOS uses that setting and ignores the default setting.

If you are using Active Directory as your LDAP server, create a user account in the Active Directory server to bind with the FXOS. This account should be given a non-expiring password.

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **LDAP** tab.

Step 3 In the **Properties** area, complete the following fields:

Name	Description
Timeout field	The length of time in seconds the system will spend trying to contact the LDAP database before it times out. Enter an integer from 1 to 60 seconds. The default value is 30 seconds. This property is required.
Attribute field	An LDAP attribute that stores the values for the user roles and locales. This property is always a name-value pair. The system queries the user record for the value that matches this attribute. Note that the `shell:roles="admin,aaa"` attribute value is required when configuring properties for LDAP providers.
Base DN field	The specific distinguished name in the LDAP hierarchy where the server should begin a search when a remote user logs in and the system attempts to get the user's DN based on their user name. The length of the base DN can be a maximum of 255 characters minus the length of *cn=$userid*, where $userid identifies the remote user attempting to access the chassis using LDAP authentication. This property is required for LDAP providers. If you do not specify a base DN on this tab, then you must specify one for each LDAP provider that you define.
Filter field	Enter the filter attribute to use with your LDAP server, for example *cn=$userid* or *sAMAccountName=$userid*. The LDAP search is restricted to those user names that match the defined filter. The filter must include *$userid*. This property is required. If you do not specify a filter on this tab then you must specify one for each LDAP provider that you define.

Step 4 Click **Save**.

What to do next

Create an LDAP provider.

Creating an LDAP Provider

Follow these steps to define and configure a LDAP provider—that is, a specific remote server providing LDAP-based AAA services for this appliance.

Note The FXOS supports a maximum of 16 LDAP providers.

Before you begin

If you are using Active Directory as your LDAP server, create a user account in the Active Directory server to bind with the FXOS. This account should be given a non-expiring password.

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **LDAP** tab.

Step 3 For each LDAP provider that you want to add:

a) In the **LDAP Providers** area, click **Add**.

b) In the **Add LDAP Provider** dialog box, complete the following fields:

Name	Description
Hostname/FQDN (or IP Address) field	The hostname or IP address of the LDAP server. If SSL is enabled, this field must exactly match a Common Name (CN) in the security certificate of the LDAP database.
Order field	The order in which the FXOS uses this provider to authenticate users. Enter an integer between 1 and 16, or enter `lowest-available` or `0` (zero) if you want the FXOS to assign the next available order based on the other providers defined in chassis manager or the FXOS CLI.
Bind DN field	The distinguished name (DN) for an LDAP database account that has read and search permissions for all objects under the base DN. The maximum supported string length is 255 ASCII characters.

Name	Description
Base DN field	The specific distinguished name in the LDAP hierarchy where the server should begin a search when a remote user logs in and the system attempts to get the user's DN based on their user name. The length of the base DN can be set to a maximum of 255 characters minus the length of CN=$userid, where $userid identifies the remote user attempting to access chassis manager or the FXOS CLI using LDAP authentication. This value is required unless a default base DN has been set on the **LDAP** tab.
Port field	The port through which chassis manager or the FXOS CLI communicates with the LDAP database. The standard port number is 389.
Enable SSL check box	If checked, encryption is required for communications with the LDAP database. If unchecked, authentication information will be sent as clear text. LDAP uses STARTTLS. This allows encrypted communication using port 389. **Note** STARTTLS operation requires the CA cert of the LDAP provider to be installed on the FXOS certificate chain.
Filter field	Enter the filter attribute to use with your LDAP server, for example *cn=$userid* or *sAMAccountName=$userid*. The LDAP search is restricted to those user names that match the defined filter. The filter must include *$userid*. This value is required unless a default filter has been set on the **LDAP** tab.
Attribute field	An LDAP attribute that stores the values for the user roles and locales. This property is always a name-value pair. The system queries the user record for the value that matches this attribute name. This value is required unless a default attribute has been set on the **LDAP** tab.
Key field	The password for the LDAP database account specified in the **Bind DN** field. You can enter any standard ASCII characters except for space, § (section sign), ? (question mark), or = (equal sign).
Confirm Key field	The LDAP database password repeated for confirmation.
Timeout field	The length of time in seconds the system will spend trying to contact the LDAP database before it times out. Enter an integer from 1 to 60 seconds, or enter 0 (zero) to use the global timeout value specified on the **LDAP** tab. The default is 30 seconds.

Name	Description
Vendor field	This selection identifies the vendor that is providing the LDAP provider or server details: • If the LDAP provider is Microsoft Active Directory, select **MS AD**. • If the LDAP provider is not Microsoft Active Directory, select **Open LDAP**. The default is **Open LDAP**.

c) Click **OK** to close the **Add LDAP Provider** dialog box.

Step 4 Click **Save**.

Step 5 (Optional) Enable the certification revocation list check:

Firepower-chassis /security/ldap/server # **set revoke-policy** *{strict | relaxed}*

Note This configuration only takes effect if the SSL connection is enabled.

Deleting an LDAP Provider

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **LDAP** tab.

Step 3 In the **LDAP Providers** area, click the **Delete** icon in the row in the table that corresponds to the LDAP Provider you want to delete.

Configuring RADIUS Providers

Configuring Properties for RADIUS Providers

The properties that you configure in this task are the default settings for all provider connections of this type. If an individual provider includes a setting for any of these properties, the FXOS uses that setting and ignores this default setting.

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **RADIUS** tab.

Step 3 In the **Properties** area, complete the following fields:

Name	Description
Timeout field	The length of time in seconds the system will spend trying to contact the RADIUS database before it times out. Enter an integer from 1 to 60 seconds. The default value is 5 seconds. This property is required.
Retries field	The number of times to retry the connection before the request is considered to have failed.

Step 4 Click **Save**.

What to do next

Create a RADIUS provider.

Creating a RADIUS Provider

Follow these steps to define and configure a RADIUS provider—that is, a specific remote server providing RADIUS-based AAA services for this appliance.

Note The FXOS supports a maximum of 16 RADIUS providers.

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **RADIUS** tab.

Step 3 For each RADIUS provider that you want to add:

a) In the **RADIUS Providers** area, click **Add**.

b) In the **Add RADIUS Provider** dialog box, complete the following fields:

Name	Description
Hostname/FQDN (or IP Address) field	The hostname or IP address of the RADIUS server.
Order field	The order in which the FXOS uses this provider to authenticate users. Enter an integer between 1 and 16, or enter `lowest-available` or `0` (zero) if you want the FXOS to assign the next available order based on the other providers defined in chassis manager or the FXOS CLI.
Key field	The SSL encryption key for the database. You can enter any standard ASCII characters except for space, § (section sign), ? (question mark), or = (equal sign).

Name	Description
Confirm Key field	The SSL encryption key repeated for confirmation.
Authorization Port field	The port through which chassis manager or the FXOS CLI communicates with the RADIUS database. The valid range is 1 to 65535. The standard port number is 1700.
Timeout field	The length of time in seconds the system will spend trying to contact the RADIUS database before it times out. Enter an integer from 1 to 60 seconds, or enter 0 (zero) to use the global timeout value specified on the **RADIUS** tab. The default is 5 seconds.
Retries field	The number of times to retry the connection before the request is considered to have failed. If desired, enter an integer between 0 and 5. If you do not specify a value, Secure Firewall chassis manager uses the value specified on the **RADIUS** tab.

c) Click **OK** to close the **Add RADIUS Provider** dialog box.

Step 4 Click **Save**.

Deleting a RADIUS Provider

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **RADIUS** tab.

Step 3 In the **RADIUS Providers** area, click the **Delete** icon in the row in the table that corresponds to the RADIUS Provider you want to delete.

Configuring TACACS+ Providers

Configuring Properties for TACACS+ Providers

The properties that you configure in this task are default settings for all provider connections of this type. If an individual provider configuration includes a setting for any of these properties, the FXOS uses that setting and ignores this default setting.

Note The FXOS chassis does not support command accounting for the Terminal Access Controller Access-Control System Plus (TACACS+) protocol.

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **TACACS** tab.

Step 3 In the **Properties** area, complete the following fields:

Name	Description
Timeout field	The length of time in seconds the system will spend trying to contact the TACACS+ database before it times out. Enter an integer from 1 to 60 seconds. The default value is 5 seconds. This property is required.

Step 4 Click **Save**.

What to do next

Create a TACACS+ provider.

Creating a TACACS+ Provider

Follow these steps to define and configure a TACACS+ provider—that is, a specific remote server providing TACACS-based AAA services for this appliance.

Note The FXOS supports a maximum of 16 TACACS+ providers.

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **TACACS** tab.

Step 3 For each TACACS+ provider that you want to add:

a) In the **TACACS Providers** area, click **Add**.

b) In the **Add TACACS Provider** dialog box, complete the following fields:

Name	Description
Hostname/FQDN (or IP Address) field	The hostname or IP address of the TACACS+ server.
Order field	The order in which the FXOS uses this provider to authenticate users. Enter an integer between 1 and 16, or enter `lowest-available` or `0` (zero) if you want the FXOS to assign the next available order based on the other providers defined in chassis manager or the FXOS CLI.

Name	Description
Key field	The SSL encryption key for the database. You can enter any standard ASCII characters except for space, § (section sign), ? (question mark), or = (equal sign).
Confirm Key field	The SSL encryption key repeated for confirmation.
Port field	The port through which chassis manager or the FXOS CLI communicates with this TACACS+ server. Enter an integer between 1 and 65535. The default port is 49.
Timeout field	The length of time in seconds the system will spend trying to contact the TACACS+ database before it times out. Enter an integer from 1 to 60 seconds, or enter 0 (zero) to use the global timeout value specified on the **TACACS+** tab. The default is 5 seconds.

c) Click **OK** to close the **Add TACACS Provider** dialog box.

Step 4 Click **Save**.

Deleting a TACACS+ Provider

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **TACACS** tab.

Step 3 In the **TACACS Providers** area, click the **Delete** icon in the row in the table that corresponds to the TACACS+ Provider you want to delete.

Configuring Single Sign-On (SSO)

A chassis manager configured for SSO presents a link for single sign-on on the Login page. Users configured for SSO access click on this link and are redirected to the IdP for authentication and authorization, rather than supplying a username and password on the chassis manager Login page. Once successfully authenticated by the IdP, SSO users are redirected back to the chassis manager web interface and logged in. All the communication between the chassis manager and the IdP to accomplish this takes place using the browser as an intermediary; as a result, the chassis manager does not require a network connection to directly access the identity provider.

The chassis manager supports SSO using any SSO provider conforming to the Security Assertion Markup Language (SAML) 2.0 open standard for authentication and authorization.

The chassis manager web interface offers configuration options for the following SSO providers:

- Okta
- OneLogin

- Azure
- PingID's PingOne for Customers cloud solution
- Cisco SSO
- Other

Configure Single Sign-On with Okta

Use these instructions at the Okta Classic UI Admin Console to create a chassis manager service provider application within Okta and assign users to that application. You should be familiar with SAML SSO concepts and the Okta admin console. This documentation does not describe all the Okta functions you need to establish a fully functional SSO org; for instance, to create users, or to import user definitions from another user management application, see the Okta documentation.

Before you begin

- Familiarize yourself with the SSO federation and its users and groups.
- Create user accounts in your Okta org if necessary.

Procedure

Step 1 From the Okta Classic UI Admin Console, create a service provider application for the chassis manager. Configure the chassis manager application with the following selections:

- Select `Web` for the **Platform**.
- Select `SAML 2.0` for the **Sign on method**.
- Provide a **Single sign on URL**.

  This is the chassis manager URL to which the browser sends information on behalf of the IdP.

  Append the string `saml/acs` to the chassis manager login URL. For example: `https://ExampleFCM/saml/acs`.
- Enable **Use this for Recipient URL and Destination URL**.
- Enter an **Audience URI (SP Entity ID)**.

  Append the string `/saml/metadata` to the login URL. For example: `https://ExampleFCM/saml/metadata`.
- For **Name ID Format**, choose `Unspecified`.

Step 2 Add a new attribute to the default Okta user profile:

- For **Data type** choose `string`.
- For **Variable name** , add string `role`.

Step 3 Assign Okta user to chassis manager.

Step 4 For user assigned to the chassis manager service provider application using this profile, assign a value to the user role attribute you have just created. You can select **admin read-only** or **read-only** based on your requirements.

Note If attribute role is not specified, chassis manager will take default role as read-only and the user will not be able to perform any edit action in chassis manager.

Step 5 Export the Identity Provider Metadata from Okta to your local system and take a note of the following values from the XML file:

- **Identity Provider Single Sign-On (SSO) URL**: Given as SingleSignOnService `Location` in Identity Provider Metadata XML file.
- **Identity Provider Issuer**: Given as `Entity ID` in Identity Provider Metadata XML file.
- **X.509 Certificate**: Given as `X509Certificate` in Identity Provider Metadata XML file.

Note These values are necessary in order to configure the Okta IDP provider in the chassis manager.

What to do next

- Enable Single Sign-On on chassis manager, see Enable/Disable Single Sign-On on Chassis Manager, on page 48.
- Configure SSO Provider in chassis manager, see Configure SSO Provider on the Chassis Manager, on page 49.

Configure Single Sign-On with OneLogin

Use these instructions at the OneLogin Admin Portal to create a chassis manager service provider application within OneLogin and assign users to that application. You should be familiar with SAML SSO concepts and the OneLogin Admin Portal. This documentation does not describe all the OneLogin functions you need to establish a fully functional SSO org; for instance, to create users, or to import user definitions from another user management application, see the OneLogin documentation.

Before you begin

- Familiarize yourself with the SSO federation and its users and groups.
- Create user accounts in your OneLogin org if necessary.

Procedure

Step 1 Create the chassis manager service provider application using the **SAML Test Connector (Advanced)** as its basis.

Step 2 Configure the application with the following settings:

- For the **Audience (Entity ID)**, append the string `/saml/metadata` to the chassis manager login URL. For example: `https://ExampleFCM/saml/metadata`.

- For **Recipient**, append the string /`saml/acs` to the chassis manager login URL. For example: `https://ExampleFCM/saml/acs`.
- For **ACS (Consumer) URL Validator**, enter an expression that OneLogin uses to confirm it is using the correct chassis manager URL. You can create a simple validator by using the ACS URL and altering it as follows:
  - Append a `^` to the beginning of the ACS URL.
  - Append a `$` to the end of the ACS URL.
  - Insert a `\` preceding every `/` and `?` within the ACS URL.

For example, for the ACS URL `https://ExampleFCM/saml/acs`, an appropriate URL validator would be `^https:\/\/ExampleFCM\/saml\/acs$`.

- For **ACS (Consumer) URL**, append the string `/saml/acs` to the chassis manager login URL. For example: `https://ExampleFCM/saml/acs`.
- For **Login URL**, append the string `/saml/acs` to the chassis manager login URL. For example: `https://ExampleFCM/saml/acs`.
- For the **SAML Initiator**, choose `Service Provider`.

Step 3 Assign OneLogin user to chassis manager.

Step 4 For user assigned to the chassis manager service provider application using this profile, assign a value to the user role attribute you have just created.

Note If attribute role is not specified, chassis manager will take default role as read-only and the user will not be able to perform any edit action in chassis manager.

Step 5 Export the SAML XML metadata from OneLogin to your local system and take a note of the following values from the XML file:

- **Identity Provider Single Sign-On (SSO) URL**: Given as `SAML 2.0 Endpoint(HTTP)` in SAML XML metadata file.
- **Identity Provider Issuer**: Given as `Issuer URL` in SAML XML metadata file.
- **X.509 Certificate**: Given as `X509Certificate` in SAML XML metadata file.

Note These values are necessary in order to configure the OneLogin IDP provider in the chassis manager.

What to do next

- Enable Single Sign-On on chassis manager, see Enable/Disable Single Sign-On on Chassis Manager, on page 48.
- Configure SSO Provider in chassis manager, see Configure SSO Provider on the Chassis Manager, on page 49.

Configure Single Sign-On with Azure AD

Use the Azure Active Directory Portal to create a chassis manager service provider application within your Azure Active Directory tenant and establish basic configuration settings.

Before you begin

- Familiarize yourself with the Azure tenant and its users and groups.
- Create user accounts in your Azure tenant org if necessary.

Procedure

Step 1 Create the chassis manager service provider application using the Azure AD SAML Toolkit as its basis.

Step 2 Configure the application with the following setttings for **Basic SAML Configuration**:

- For the **Identifier (Entity ID)** append the string `/saml/metadata` to the chassis manager login URL. For example: `https://ExampleFCM/saml/metadata`.
- For the **Reply URL (Assertion Consumer Service URL)** append the string `/saml/acs` to the chassis manager login URL. For example: `https://ExampleFCM/saml/acs`.
- For the **Sign on URL** append the string `/saml/acs` to the chassis manager login URL. For example: `https://ExampleFCM/saml/acs`.

Step 3 Edit the **Unique User Identifier Name (Name ID)** claim for the application to force the username for sign-on at the chassis manager to be the email address associated with the user account:

- For **Source** choose `Attribute`.
- For **Source attribute**: Choose `user.mail`.

Step 4 Generate a certificate to secure SSO on the chassis manager. Use the following options for the certificate:

- Select Sign SAML Response and Assertion for the Signing Option.
- Select SHA-256 for the Signing Algorithm.

Step 5 Download the Base-64 version of the certificate to your local computer; you need add the contents as **X.509 Certificate** when you configure Azure SSO at the chassis manager web interface.

Step 6 In the SAML-based Sign-on information for the application, note the following values:

- **Login URL**:
- **Azure AD Identifier**

You will need these values when you configure Azure SSO at the chassis manager web interface.

Note The identity provider's single sign-on URL is the Login URL, and the identity provider's issuer is the Azure AD Identifier.

Step 7 Assign Azure user to chassis manager.

Step 8 For user assigned to the chassis manager service provider application using this profile, assign a value to the user role attribute you have just created.

Note If attribute role is not specified, chassis manager will take default role as read-only and the user will not be able to perform any edit action in chassis manager.

What to do next

- Enable Single Sign-On on chassis manager, see Enable/Disable Single Sign-On on Chassis Manager, on page 48.
- Configure SSO Provider in chassis manager, see Configure SSO Provider on the Chassis Manager, on page 49.

Configure Single Sign-On with PingID

Use the PingOne for Customers Administrator Console to create a chassis manager service provider application within your PingOne for Customers environment and establish basic configuration settings. This documentation does not describe all the PingOne for Customers functions you need to establish a fully functional SSO environment; for instance, to create users see the PingOne for Customers documentation.

Before you begin

- Familiarize yourself with your PingOne for Customers environment and its users.
- Create additional users if necessary.

Procedure

Step 1 Use the PingOne for Customer Administrator Console to create the application in your environment using these settings:

- Choose the **Web App** application type.
- Choose the **SAML** connection type.

Step 2 Configure the application with the following settings for the SAML Connection:

- For the **ACS URL**, append the string `/sam/acs` to the chassis manager login URL. For example: `https://ExampleFCM/saml/acs`.
- For the **Signing Certificate**, choose Sign Assertion & Response.
- For the **Signing Algorithm** choose RSA_SHA256.
- For the **Entity ID**, append the string `/saml/metadata` to the chassis manager login URL. For example: `https://ExampleFCM/saml/metadata`.
- For the **SLO Binding** select HTTP POST.
- For the **Assertion Validity Duration** enter 300.

Step 3 In the SAMLConnection information for the application, note the following values:

- **Single Sign-On Service**
- **Issuer ID**

You will need these values when you configure SSO using PingID's PingOne for Customers product at the chassis manager web interface.

Step 4 For **SAML ATTRIBUTES**, make the following selections for a single required attribute:

- **PINGONE USER ATTRIBUTE:** `Email Address`
- **APPLICATION ATTRIBUTE:** `saml_subject`

Step 5 Download the signing certificate in X509 PEM (`.crt`) format and save it to your local computer.

You will need these cert when you configure SSO using PingID's PingOne for Customers product at the chassis manager web interface.

Step 6 Enable the application.

What to do next

- Enable Single Sign-On on chassis manager, see Enable/Disable Single Sign-On on Chassis Manager, on page 48.
- Configure SSO Provider in chassis manager, see Configure SSO Provider on the Chassis Manager, on page 49.

Configure Single Sign-On with Cisco SSO

Duo Single Sign-On is a cloud-hosted single sign-on solution (SSO) solution which can act as a SAML 2.0 identity provider that secures access to chassis manager with your existing directory credentials. Duo Single Sign-On allows you to use either Active Directory domains and SAML Identity Provider as a first-factor authentication source. For SSO, Duo uses SAML authentication from chassis manager to an identity provider. You can configure your SAML 2.0 identity provider and chassis manager on Duo using the below steps.

For configuring SSO using Active Directory, see Single Sign-On using Active Directory.

Before you begin

- Familiarize yourself with the SSO federation and its users and groups.
- A Duo Admin account with the **Owner** role to enable the feature.
- Active Directory or a SAML identity provider that can be used as your primary authentication source for Duo Single Sign-On.

Procedure

Step 1 On the "Single Sign-On Configuration" page scroll down to **Configure your SAML Identity Provider**. This is the Duo Single Sign-On metadata information you need to provide to your SAML identity provider application to configure Duo Single Sign-On as a service provider.

Step 2 In the "SAML Certificates" section of the properties page of your SAML provider application, click **Download** next to **Certificate (Base64)**. You will need this certificate file.

Step 3 Configure the chassis manager on the Service Provider page with the following settings:

- For the **Entity ID**, append the string `/saml/metadata` to the chassis manager login URL. For example: `https://ExampleFCM/saml/metadata`.
- For **Assertion Consumer Service (ACS) URL**, append the string `/saml/acs` to the login URL. For examchassis manager: `https://ExampleFCM/saml/acs`.
- For **Service Provider Login URL**, append the string `/saml/acs` to the chassis manager login URL. For example: `https://ExampleFCM/saml/acs`.
- For **Certificate**, upload the certificate downloaded from your SAML service provider application.

Step 4 Click on **Download Certificate**. This is the X.509 Certificate that you need to add while configuring Cisco SSO on chassis manager.

For **Identity Provider Single Sign-On (SSO) URL** and **Identity Provider Issuer**, use the details from Duo Single Sign-On metadata information page.

What to do next

- Enable Single Sign-On on chassis manager, see Enable/Disable Single Sign-On on Chassis Manager, on page 48.
- Configure SSO Provider in chassis manager, see Configure SSO Provider on the Chassis Manager, on page 49.

Configure Single Sign-On with Any SAML 2.0-Compliant SSO Provider

Generally SSO providers require that you configure a service provider application at the IdP for each federated application. All IdPs that support SAML 2.0 SSO need the same configuration information for service provider applications, but some IdP's automatically generate some configuration settings for you, while others require that you configure all settings yourself.

Before you begin

- Familiarize yourself with the SSO federation and its users and groups.
- Confirm your IdP account has the necessary permissions to perform this task.
- Create user accounts and/or groups in your SSO federation if necessary.

Procedure

Step 1 Create a new service provider application at the IdP.

Step 2 Configure values required by the IdP. Be sure to include the fields listed below, required to support SAML 2.0 SSO functionality with the chassis manager. (Because different SSO service providers use different terminology for SAML concepts, this list provides alternate names for these fields to help you find the right settings in the IdP application.):

- Service Provider Entity ID, Service Provider Identifier, Audience URI: A globally unique name for the service provider (the chassis manager), formatted as a URL. To create this, append the string `/saml/metadata` to the chassis manager login URL, such as `https://ExampleFCC/saml/metadata`.
- Single Sign on URL, Recipient URL, Assertion Consumer Service URL: The service provider (chassis manager) address to which the browser sends information on behalf of the IdP. To create this, append the string `saml/acs` to the chassis manager login URL, such as `https://ExampleFCM/saml/acs`.
- X.509 Certificate: Certificate to secure communications between the chassis manager and the IdP. Some IdP's may automatically generate the certificate, and some may require that you explicitly generate it using the IDP interface.

Step 3 (Optional if you are assigning groups to the application) Assign individual users to the chassis manager application.

Step 4 At the IdP, create or designate an attribute to be sent to the chassis manager to contain role mapping information for each user sign-in. This may be a user attribute or a different attribute that obtains its value from a source such as user or group definitions maintained by the IdP or a third party user management application.

Step 5 (Optional) Some IdP's provide the ability to generate a SAML XML metadata file containing the information you have configured in this task formatted to comply with SAML 2.0 standards. You can take a note of the required values and use them while configuring the IDP on the chassis manager

What to do next

- Enable Single Sign-On on chassis manager, see Enable/Disable Single Sign-On on Chassis Manager, on page 48.
- Configure SSO Provider in chassis manager, see Configure SSO Provider on the Chassis Manager, on page 49.

Enable/Disable Single Sign-On on Chassis Manager

Before you begin

- At the SAML SSO management application, configure a service provider application for the chassis manager and assign users or groups to the service provider application:
  - To configure a chassis manager service provider application for Okta, see Configure Single Sign-On with Okta, on page 41.
  - To configure a chassis manager service provider application for OneLogin, see Configure Single Sign-On with OneLogin, on page 42.

- To configure a chassis manager service provider application for Azure, see Configure Single Sign-On with Azure AD, on page 44.
- To configure a chassis manager service provider application for PingID's PingOne for Customers cloud solution, see Configure Single Sign-On with PingID, on page 45.
- To configure a chassis manager service provider application for any SAML 2.0-compliant SSO provider, see Configure Single Sign-On with Any SAML 2.0-Compliant SSO Provider, on page 47.

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **Single Sign-On (SSO)** tab.

Step 3 To enable Single Sign-On (SSO) access to the chassis, check the **Enable Single Sign-On (SSO)** check box. To disable Single Sign-On (SSO) access, uncheck the **Enable Single Sign-On (SSO)** check box.

Step 4 Click **Save**.

What to do next

Configure an SSO provider.

Configure SSO Provider on the Chassis Manager

Before you begin

- Create a chassis manager service provider application at the SSO service provider and retrieve the values for configuring the service provider in chassis manager.
- Enable single sign-on; see Enable/Disable Single Sign-On on Chassis Manager, on page 48.

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **Single Sign-On (SSO)** tab.

Step 3 (Optional) In the **Configure SSO Provider** area, click **Add**.

Step 4 (Optional) In the **SSO Provider** area, click the **Edit** icon available on the listed SSO Provider.

Step 5 In the **Configure SSO Provider** the, do the following :

a) Select the SSO provider from the **Choose the SAML Provider** drop-down list.

Note You can select the SSO service provider on which you have already created the chassis service provider application.

b) Enter the values you retrieved from the SSO service provider in the following fields:

- **Identity Provider Single Sign-On URL**

- **Identity Provider Issuer**
- **X.509 Certificates**

Note You can use the values retrieved from the SSO service provider or XML metadata file.

Step 6 Review the configuration parameters and click **Save**.

Step 7 Click **Test Configuration**. If the system displays an error message, review the SSO configuration for the chassis as well as the SSO service provider application configuration, correct any errors, and try again.

Deleting an SSO Provider

Procedure

Step 1 Choose **Platform Settings** > **AAA**.

Step 2 Click the **Single Sign-On (SSO)** tab.

Step 3 In the **SSO Providers** area, click the **Delete** icon in the table that corresponds to the LDAP Provider you want to delete.

Step 4 In the **Confirm** dialog box, click **Yes** to delete the SSO provider.

Configuring Syslog

System logging is a method of collecting messages from devices to a server running a syslog daemon. Logging to a central syslog server helps in aggregation of logs and alerts. A syslog service accepts messages and stores them in files, or prints them according to a simple configuration file. This form of logging provides protected long-term storage for logs. Logs are useful both in routine troubleshooting and in incident handling.

Procedure

Step 1 Choose **Platform Settings** > **Syslog**.

Step 2 Configure Local Destinations:

a) Click the **Local Destinations** tab.

b) On the **Local Destinations** tab, complete the following fields:

Name	Description
Console Section	
Admin State field	Whether the chassis displays syslog messages on the console. Check the **Enable** check box if you want to have syslog messages displayed on the console as well as added to the log. If the **Enable** check box is unchecked, syslog messages are added to the log but are not displayed on the console.

Name	Description
Level field	If you checked the **Enable** check box for **Console - Admin State**, select the lowest message level that you want displayed on the console. The chassis displays that level and above on the console. This can be one of the following: • **Emergencies** • **Alerts** • **Critical**
Monitor Section	
Admin State field	Whether the chassis displays syslog messages on the monitor. Check the **Enable** check box if you want to have syslog messages displayed on the monitor as well as added to the log. If the **Enable** check box is unchecked, syslog messages are added to the log but are not displayed on the monitor.
Level drop-down list	If you checked the **Enable** check box for **Monitor - Admin State**, select the lowest message level that you want displayed on the monitor. The system displays that level and above on the monitor. This can be one of the following: • **Emergencies** • **Alerts** • **Critical** • **Errors** • **Warnings** • **Notifications** • **Information** • **Debugging**

c) Click **Save**.

Step 3 Configure Remote Destinations:

a) Click the **Remote Destinations** tab.

b) On the **Remote Destinations** tab, complete the following fields for up to three external logs that can store messages generated by the chassis:

By sending syslog messages to a remote destination, you can archive messages according to the available disk space on the external syslog server, and manipulate logging data after it is saved. For example, you could specify actions to be executed when certain types of syslog messages are logged, extract data from the log and save the records to another file for reporting, or track statistics using a site-specific script.

Name	Description
Admin State field	Check the **Enable** check box if you want to have syslog messages stored in a remote log file.
Level drop-down list	Select the lowest message level that you want the system to store. The system stores that level and above in the remote file. This can be one of the following: • **Emergencies** • **Alerts** • **Critical** • **Errors** • **Warnings** • **Notifications** • **Information** • **Debugging**
Hostname/IP Address field	The hostname or IP address on which the remote log file resides. **Note** You must configure a DNS server if you use a hostname rather than an IP address.
Facility drop-down list	Choose a system log facility for syslog servers to use as a basis to file messages. This can be one of the following: • **Local0** • **Local1** • **Local2** • **Local3** • **Local4** • **Local5** • **Local6** • **Local7**

c) Click **Save**.

Step 4 Configure Local Sources:

a) Click the **Local Sources** tab.

b) On the **Local Sources** tab, complete the following fields:

Name	Description
Faults Admin State field	Whether system fault logging is enabled or not. If the **Enable** check box is checked, the chassis logs all system faults.
Audits Admin State field	Whether audit logging is enabled or not. If the **Enable** check box is checked, the chassis logs all audit log events.
Events Admin State field	Whether system event logging is enabled or not. If the **Enable** check box is checked, the chassis logs all system events.

c) Click **Save**.

Configuring DNS Servers

You need to specify a DNS server if the system requires resolution of host names to IP addresses. For example, you cannot use a name such as www.cisco.com when you are configuring a setting on the chassis if you do not configure a DNS server. You would need to use the IP address of the server, which can be either an IPv4 or an IPv6 address. You can configure up to four DNS servers.

Note When you configure multiple DNS servers, the system searches for the servers only in any random order. If a local management command requires DNS server lookup, it can only search for three DNS servers in random order.

Procedure

Step 1 Choose **Platform Settings** > **DNS**.

Step 2 Check the **Enable DNS Server** check box.

Step 3 For each DNS server that you want to add, up to a maximum of four, enter the IP address of the DNS server in the **DNS Server** field and click **Add**.

Step 4 Click **Save**.

Enable FIPS Mode

Perform these steps to enable FIPS mode on your Firepower 4100/9300 chassis.

Procedure

Step 1 Log into the Firepower 4100/9300 chassis as an admin user.

Step 2 Choose **Platform Settings** to open the Platform Settings window.

Step 3 Choose **FIPS/CC mode** to open the FIPS and Common Criteria window.

Step 4 Check the **Enable** checkbox for FIPS.

Step 5 Click **Save** to save the configuration.

Step 6 Follow the prompt to reboot the system.

When the FIPS Mode is enabled, it limits the key sizes and the algorithms allowed. The MIO uses CiscoSSL and the FIPS Object Module (FOM) for its cryptographic needs. It makes FIPS validation easier compared to ASA's proprietary cryptographic library implementation and HW acceleration.

What to do next

Prior to FXOS release 2.0.1, the existing SSH host key created during first-time setup of a device was hard coded to 1024 bits. To comply with FIPS and Common Criteria certification requirements, you must destroy this old host key and generate a new one using the procedure detailed in Generate theSSH Host Key. If you do not perform these additional steps, you will not be able to connect to the Supervisor using SSH after the device has rebooted with FIPS mode enabled. If you performed initial setup using FXOS 2.0.1 or later, you do not have to generate a new host key.

Enable Common Criteria Mode

Perform these steps to enable Common Criteria mode on your Firepower 4100/9300 chassis.

Procedure

Step 1 Log into the Firepower 4100/9300 chassis as an admin user.

Step 2 Choose **Platform Settings** to open the Platform Settings window.

Step 3 Choose **FIPS/CC mode** to open the FIPS and Common Criteria window.

Step 4 Check the **Enable** checkbox for Common Criteria.

Step 5 Click **Save** to save the configuration.

Step 6 Follow the prompt to reboot the system.

Common Criteria is an international standard for computer security. CC focuses on certificates, auditing, logging, passwords, TLS, SSH, etc. It essentially assumes FIPS compliance. Similar to FIPS, Cisco contracts with NIST accredited lab vendors to perform testing and submission to NIAP.

When the CC Mode is enabled, it limits the list of algorithms, cipher suites, and features that are needed to be supported. The MIO is evaluated against the Network Device Collaborative Protection Profile (NDcPP). CiscoSSL can only enforce part of the requirements most of which are covered in the CC compliance guide.

What to do next

Prior to FXOS release 2.0.1, the existing SSH host key created during first-time setup of a device was hard coded to 1024 bits. To comply with FIPS and Common Criteria certification requirements, you must destroy this old host key and generate a new one using the procedure detailed in Generate the SSH Host Key. If you do not perform these additional steps, you will not be able to connect to the Supervisor using SSH after the

device has rebooted with Common Criteria mode enabled. If you performed initial setup using FXOS 2.0.1 or later, you do not have to generate a new host key.

Configure the IP Access List

By default, the Firepower 4100/9300 chassis denies all access to the local web server. You must configure your IP Access List with a list of allowed services for each of your IP blocks.

The IP Access List supports the following protocols:

- HTTPS
- SNMP
- SSH

For each block of IP addresses (v4 or v6), up to 100 different subnets can be configured for each service. A subnet of 0 and a prefix of 0 allows unrestricted access to a service.

Procedure

Step 1 Log into the Firepower 4100/9300 chassis as an admin user.

Step 2 Choose **Platform Settings** to open the Platform Settings page.

Step 3 Select **Access List** to open the Access List area.

Step 4 In this area, you can view, add, and delete the IPv4 and IPv6 addresses listed in your IP Access List.

To add an IPv4 block, you must enter a valid IPv4 IP address, a prefix [0-32] length, and select a protocol.

To add an IPv6 block, you must enter a valid IPv6 IP address, a prefix [0-128] length, and select a protocol

Add a MAC Pool Prefix and View MAC Addresses for Container Instance Interfaces

The FXOS chassis automatically generates MAC addresses for container instance interfaces, and guarantees that a shared interface in each instance uses a unique MAC address. The FXOS chassis generates the MAC address using the following format:

A2*xx.yyzz.zzzz*

Where *xx.yy* is a user-defined prefix or a system-defined prefix, and *zz.zzzz* is an internal counter generated by the chassis. The system-defined prefix matches the lower 2 bytes of the first MAC address in the burned-in MAC address pool that is programmed into the IDPROM. Use **connect fxos**, then **show module** to view the MAC address pool. For example, if the range of MAC addresses shown for module 1 is b0aa.772f.f0b0 to b0aa.772f.f0bf, then the system prefix will be f0b0.

See Automatic MAC Addresses for Container Instance Interfaces for more information.

This procedure describes how to view the MAC addresses and how to optionally define the prefix used in generation.

Note If you change the MAC address prefix after you deploy logical devices, you may experience traffic interruption.

Procedure

Step 1 Choose **Platform Settings** > **MAC Pool**.

This page shows generated MAC addresses along with the container instance and interface using the MAC address.

Step 2 (Optional) Add a MAC address prefix used in generating the MAC addresses.

a) Click **Add Prefix**.

The **Set the Prefix for the MAC Pool** dialogue box appears.

a) Enter a decimal value between 1 and 65535. This prefix is converted to a four-digit hexadecimal number, and used as part of the MAC address.

For an example of how the prefix is used, if you set a prefix of 77, then the chassis converts 77 into the hexadecimal value 004D (*yyxx*). When used in the MAC address, the prefix is reversed (*xxyy*) to match the chassis native form:

A2**4D.00***zz.zzzz*

For a prefix of 1009 (03F1), the MAC address is:

A2**F1.03***zz.zzzz*

b) Click **OK**.

New MAC addresses using the prefix are generated and assigned. The current prefix and the resulting hex value display above the table.

Add a Resource Profile for Container Instances

To specify resource usage per container instance, create one or more resource profiles. When you deploy the logical device/application instance, you specify the resource profile that you want to use. The resource profile sets the number of CPU cores; RAM is dynamically allocated according to the number of cores, and disk space is set to 40 GB per instance.

- The minimum number of cores is 6.

Note Instances with a smaller number of cores might experience relatively higher CPU utilization than those with larger numbers of cores. Instances with a smaller number of cores are more sensitive to traffic load changes. If you experience traffic drops, try assigning more cores.

- You can assign cores as an even number (6, 8, 10, 12, 14 etc.) up to the maximum.
- The maximum number of cores available depends on the security module/chassis model; see Requirements and Prerequisites for Container Instances.

The chassis includes a default resource profile called "Default-Small," which includes the minimum number of cores. You can change the definition of this profile, and even delete it if it is not in use. Note that this profile is created when the chassis reloads and no other profile exists on the system.

You cannot change the resource profile settings if it is currently in use. You must disable any instances that use it, then change the resource profile, and finally reenable the instance. If you resize instances in an established High Availability pair or cluster, then you should make all members the same size as soon as possible.

If you change the resource profile settings after you add the threat defense instance to the management center, then update the inventory for each unit on the management center **Devices** > **Device Management** > **Device** > **System** > **Inventory** dialog box.

Procedure

Step 1 Choose **Platform Settings** > **Resource Profiles** , and click **Add**.

The **Add Resource Profile** dialog box appears.

Step 2 Set the following paramters.

- **Name**—Sets the name of the profile between 1 and 64 characters. Note that you cannot change the name of this profile after you add it.
- **Description**—Sets the description of the profile up to 510 characters.
- **Number of Cores**—Sets the number of cores for the profile, between 6 and the maximum, depending on your chassis, as an even number.

Step 3 Click **OK**.

Configure a Network Control Policy

To permit the discovery of non-Cisco devices, FXOS supports the *Link Layer Discovery Protocol (LLDP)*, a vendor-neutral device discovery protocol that is defined in the IEEE 802.1ab standard. LLDP allows network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other.

LLDP is a one-way protocol that transmits information about the capabilities and current status of a device and its interfaces. LLDP devices use the protocol to solicit information only from other LLDP devices.

To enable this functionality on your FXOS chassis, you can configure a network control policy, which specifies LLDP transmission and receiving behavior. Once a network control policy is created, it needs to be assigned to an interface. You can enable LLDP on any front interface, including fixed ports, EPM ports, port channels, and break out ports.

Note

- LLDP is not configurable on dedicated management ports.
- Internal backplane ports that connect to the blade have LLDP enabled by default, with no option to disable. All other ports have LLDP disabled by default.

Procedure

Step 1 Choose **Platform Settings** > **Network Control Policy**.

Step 2 Click **Add**.

Step 3 In the Network Control Policy dialog box, edit the following fields:

Name	Description
Name field	A unique name for the Network Control Policy.
LLDP recieve checkbox	Enables FXOS to receive LLDP packets.
LLDP transmit checkbox	Enables FXOS to transmit LLDP packets.
Description field	A description for the Network Control Policy.

Step 4 Click **Save**. After creating the Network Control Policy, you must assign it to an interface. For steps to edit and configure an interface with a Network Control Policy, see Configure a Physical Interface.

Configure the Chassis URL

You can specify a management URL so that you can easily open chassis manager for an threat defense instance directly from management center. If you do not specify a chassis management URL, the chassis name is used instead.

If you change the chassis URL settings after you add the threat defense instance to the management center, then update the inventory for each unit on the **Devices** > **Device Management** > **Device** > **System** > **Inventory** dialog box.

Procedure

Step 1 Choose **Platform Settings** > **Chassis URL**.

Step 2 Set the following parameters.

- **Chassis Name**—Sets the name of the chassis between 1 and 60 characters.
- **Chassis URL**—Sets the URL that management center should use to connect to an threat defense instance within chassis manager. The URL must start with https://. If you do not specify a chassis management URL, the chassis name is used instead.

Step 3 Click **Update**.