Platform Settings

Setting the Date and Time

Use the CLI commands described below 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 Firepower 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 FMC, but note that you cannot use FMC 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 Connect to the FXOS CLI (see Accessing the FXOS CLI).

Step 2 To view the configured time zone:

Firepower-chassis# **show timezone**

Step 3 To view the configured date and time:

Firepower-chassis# **show clock**

Example

The following example shows how to display the configured time zone and current system date and time:

```
Firepower-chassis# show timezone
Timezone: America/Chicago
Firepower-chassis# show clock
Thu Jun  2 12:40:42 CDT 2016
Firepower-chassis#
```

Setting the Time Zone

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 time zone:

Firepower-chassis /system/services # **set timezone**

At this point, you are prompted to enter a number corresponding to your continent, country, and time zone region. Enter the appropriate information at each prompt.

When you have finished specifying the location information, you are prompted to confirm that the correct time zone information is being set. Enter **1** (yes) to confirm, or **2** (no) to cancel the operation.

Step 4 To view the configured time zone:

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

Firepower-chassis# **show timezone**

Example

The following example configures the time zone to the Pacific time zone region, commits the transaction, and displays the configured time zone:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # set timezone
Please identify a location so that time zone rules can be set correctly.
Please select a continent or ocean.
1) Africa            4) Arctic Ocean     7) Australia       10) Pacific Ocean
2) Americas          5) Asia             8) Europe
3) Antarctica        6) Atlantic Ocean   9) Indian Ocean
#? 2
Please select a country.
 1) Anguilla                 28) Haiti
 2) Antigua & Barbuda        29) Honduras
 3) Argentina                30) Jamaica
 4) Aruba                    31) Martinique
 5) Bahamas                  32) Mexico
 6) Barbados                 33) Montserrat
 7) Belize                   34) Nicaragua
 8) Bolivia                  35) Panama
 9) Brazil                   36) Paraguay
10) Canada                   37) Peru
11) Caribbean Netherlands    38) Puerto Rico
12) Cayman Islands           39) St Barthelemy
13) Chile                    40) St Kitts & Nevis
14) Colombia                 41) St Lucia
15) Costa Rica               42) St Maarten (Dutch part)
16) Cuba                     43) St Martin (French part)
17) Curacao                  44) St Pierre & Miquelon
18) Dominica                 45) St Vincent
19) Dominican Republic       46) Suriname
20) Ecuador                  47) Trinidad & Tobago
21) El Salvador              48) Turks & Caicos Is
22) French Guiana            49) United States
23) Greenland                50) Uruguay
24) Grenada                  51) Venezuela
25) Guadeloupe               52) Virgin Islands (UK)
26) Guatemala                53) Virgin Islands (US)
27) Guyana
#? 49
Please select one of the following time zone regions.
 1) Eastern Time
 2) Eastern Time - Michigan - most locations
 3) Eastern Time - Kentucky - Louisville area
 4) Eastern Time - Kentucky - Wayne County
 5) Eastern Time - Indiana - most locations
 6) Eastern Time - Indiana - Daviess, Dubois, Knox & Martin Counties
 7) Eastern Time - Indiana - Pulaski County
 8) Eastern Time - Indiana - Crawford County
 9) Eastern Time - Indiana - Pike County
10) Eastern Time - Indiana - Switzerland County
11) Central Time
12) Central Time - Indiana - Perry County
13) Central Time - Indiana - Starke County
14) Central Time - Michigan - Dickinson, Gogebic, Iron & Menominee Counties
15) Central Time - North Dakota - Oliver County
16) Central Time - North Dakota - Morton County (except Mandan area)
17) Central Time - North Dakota - Mercer County
18) Mountain Time
19) Mountain Time - south Idaho & east Oregon
20) Mountain Standard Time - Arizona (except Navajo)
21) Pacific Time
22) Pacific Standard Time - Annette Island, Alaska
23) Alaska Time
24) Alaska Time - Alaska panhandle
25) Alaska Time - southeast Alaska panhandle
26) Alaska Time - Alaska panhandle neck
27) Alaska Time - west Alaska
28) Aleutian Islands
29) Hawaii
#? 21

The following information has been given:

        United States
        Pacific Time

Therefore timezone 'America/Los_Angeles' will be set.
Local time is now:      Wed Jun 24 07:39:25 PDT 2015.
Universal Time is now:  Wed Jun 24 14:39:25 UTC 2015.
Is the above information OK?
1) Yes
2) No
#? 1
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services # top
Firepower-chassis# show timezone
Timezone: America/Los_Angeles (Pacific Time)
Firepower-chassis#
```

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, the application instance cannot sync to the NTP server on the FXOS chassis. Each time a NTP client syncs to a NTP server, the stratum value increases by one.

  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).

Before you begin

If you use a hostname for the NTP server, you must configure a DNS server. See Configuring DNS Servers, on page 54.

Procedure

Step 1 Enter system mode:

Firepower-chassis# **scope system**

Step 2 Enter system services mode:

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

Step 3 Configure the system to use the NTP server with the specified hostname, IPv4, or IPv6 address:

Firepower-chassis /system/services # **create ntp-server** *{hostname* | *ip-addr* | *ip6-addr}*

Step 4 (Optional) Configure NTP authentication.

Only SHA1 is supported for NTP server authentication. 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.

a) Set the SHA1 Key ID.

   set ntp-sha1-key-id *key_id*

b) Set the SHA1 Key String.

   set ntp-sha1-key-string

   You are prompted for the key string.

c) Exit ntp-server mode.

   exit

d) Enable NTP authentication.

   enable ntp-authentication

Example:

```
firepower /system/services/ntp-server* # set ntp-sha1-key-string 11
firepower /system/services/ntp-server* # set ntp-sha1-key-string
NTP SHA-1 key string: 7092334a7809ab9873124c08123df9097097fe72
firepower /system/services/ntp-server* # exit
firepower /system/services* # enable authentication
```

Step 5 Commit the transaction to the system configuration:

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

Step 6 To view the synchronization status for all configured NTP servers:

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

Step 7 To view the synchronization status for a specific NTP server:

Firepower-chassis /system/services # **scope ntp-server** *{hostname | ip-addr | ip6-addr}*

Firepower-chassis /system/services/ntp-server # **show detail**

Example

The following example configures an NTP server with the IP address 192.168.200.101 and commits the transaction:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # create ntp-server 192.168.200.101
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

The following example configures an NTP server with the IPv6 address 4001::6 and commits the transaction:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # create ntp-server 4001::6
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

Deleting an NTP Server

Procedure

Step 1 Enter system mode:

Firepower-chassis# **scope system**

Step 2 Enter system services mode:

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

Step 3 Delete the NTP server with the specified hostname, IPv4, or IPv6 address:

Firepower-chassis /system/services # **delete ntp-server** *{hostname | ip-addr | ip6-addr}*

Step 4 Commit the transaction to the system configuration:

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

Example

The following example deletes the NTP server with the IP address 192.168.200.101 and commits the transaction:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # delete ntp-server 192.168.200.101
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

The following example deletes the NTP server with the IPv6 address 4001::6 and commits the transaction:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # delete ntp-server 4001::6
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

Setting the Date and Time Manually

This section describes how to set the date and time manually on the chassis. System clock modifications take effect on the chassis immediately. 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).

Note If the system clock is currently being synchronized with an NTP server, you will not be able to set the date and time manually.

Procedure

Step 1 Enter system mode:

Firepower-chassis# **scope system**

Step 2 Enter system services mode:

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

Step 3 Configure the system clock:

Firepower-chassis /system/services # **set clock** *month day year hour min sec*

For month, use the first three digits of the month. Hours must be entered using the 24-hour format, where 7 pm would be entered as 19.

System clock modifications take effect immediately. You do not need to commit the buffer.

Example

The following example configures the system clock:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # set clock jun 24 2015 15 27 00
Firepower-chassis /system/services #
```

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 Enter system mode:

Firepower-chassis # **scope system**

Step 2 Enter system services mode:

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

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

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

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

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

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

Step 4 Configure encryption algorithms for the server:

Firepower-chassis /system/services # **set ssh-server encrypt-algorithm** *encrypt_algorithm*

Example:

```
Firepower /system/services # set ssh-server encrypt-algorithm ?
  3des-cbc    3des Cbc
  aes128-cbc  Aes128 Cbc
  aes128-ctr  Aes128 Ctr
  aes192-cbc  Aes192 Cbc
  aes192-ctr  Aes192 Ctr
  aes256-cbc  Aes256 Cbc
  aes256-ctr  Aes256 Ctr
```

Example:

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 5 Configure the server Diffie-Hellman (DH) key exchange algorithms:

Firepower-chassis /system/services # **set ssh-server kex-algorithm**

Example:

```
Firepower /system/services # set ssh-server kex-algorithm
  diffie-hellman-group1-sha1   Diffie Hellman Group1 Sha1
  diffie-hellman-group14-sha1  Diffie Hellman Group14 Sha1
```

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 6 Set the server mac algorithms:

Firepower-chassis /system/services # **set ssh-server mac-algorithm**

Example:

```
Firepower /system/services # set ssh-server mac-algorithm
  hmac-sha1      Hmac Sha1
  hmac-sha1-160  Hmac Sha1 160
  hmac-sha1-96   Hmac Sha1 96
  hmac-sha2-256  Hmac Sha2 256
  hmac-sha2-512  Hmac Sha2 512
```

Step 7 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.

Firepower-chassis /system/services # **set ssh-server host-key rsa** *modulus_value*

Example:

```
Firepower /system/services # set ssh-server host-key rsa ?
<1024-2048>  Enter number of bits (in multiples of 8)
Firepower /system/services # set ssh-server host-key rsa 2048
```

Step 8 For the server volume rekey limit, set the amount of traffic in KB allowed over the connection before FXOS disconnects from the session:

Firepower-chassis /system/services # **set ssh-server rekey-limit volume** *KB_of_Traffic*

Example:

```
Firepower /system/services # set /system/services # set ssh-server rekey-limit volume ?
  100-4194303  Max volume limit in KB
```

Step 9 For the server time rekey limit, set the number of minutes that an SSH session can be idle before FXOS disconnects the session:

Firepower-chassis /system/services # **set ssh-server rekey-limit time** *minutes*

Example:

```
Firepower /system/services # set /system/services # set ssh-server rekey-limit time ?
   10-1440  Max time limit in Minutes
```

Step 10 Commit the transaction to the system configuration:

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

Step 11 Configure strict host keycheck, to control SSH host key checking:

Firepower /system/services # **ssh-client stricthostkeycheck** *enable/disable/prompt*

Example:

```
Firepower /system/services # set ssh-client stricthostkeycheck enable
```

- **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 Configure encryption algorithms for the client:

Firepower-chassis /system/services # **set ssh-client encrypt-algorithm** *encrypt_algorithm*

Example:

```
Firepower /system/services # set ssh-client encrypt-algorithm ?
  3des-cbc    3des Cbc
  aes128-cbc  Aes128 Cbc
  aes128-ctr  Aes128 Ctr
  aes192-cbc  Aes192 Cbc
  aes192-ctr  Aes192 Ctr
  aes256-cbc  Aes256 Cbc
  aes256-ctr  Aes256 Ctr
```

Note

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

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

Step 13 Configure the client Diffie-Hellman (DH) key exchange algorithms:

Firepower-chassis /system/services # **set ssh-client kex-algorithm**

Example:

```
Firepower /system/services # set ssh-client kex-algorithm
  curve25519-sha256               curve25519-sha256
  curve25519-sha256_libssh_org    curve25519-sha256@libssh.org
  diffie-hellman-group14-sha1     diffie-hellman-group14-sha1
  diffie-hellman-group14-sha256   diffie-hellman-group14-sha256
  ecdh-sha2-nistp256              ecdh-sha2-nistp256
  ecdh-sha2-nistp384              ecdh-sha2-nistp384
  ecdh-sha2-nistp521              ecdh-sha2-nistp521
```

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.

Step 14 Set the client mac algorithms:

Firepower-chassis /system/services # **set ssh-client mac-algorithm**

Example:

```
Firepower /system/services # set ssh-client mac-algorithm
  hmac-sha1      Hmac Sha1
  hmac-sha1-160  Hmac Sha1 160
  hmac-sha1-96   Hmac Sha1 96
  hmac-sha2-256  Hmac Sha2 256
  hmac-sha2-512  Hmac Sha2 512
```

Step 15 For the client 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.

Firepower-chassis /system/services # **set ssh-client host-key rsa** *modulus_value*

Example:

```
Firepower /system/services # set ssh-client host-key rsa ?
<1024-2048>  Enter number of bits (in multiples of 8)
Firepower /system/services # set ssh-client host-key rsa 2048
```

Step 16 For the client volume rekey limit, set the amount of traffic in KB allowed over the connection before FXOS disconnects from the session:

Firepower-chassis /system/services # **set ssh-client rekey-limit volume** *KB_of_Traffic*

Example:

```
Firepower /system/services # set /system/services # set ssh-client rekey-limit volume ?
  100-4194303  Max volume limit in KB
```

Step 17 For the client time rekey limit, set the number of minutes that an SSH session can be idle before FXOS disconnects the session:

Firepower-chassis /system/services # **set ssh-client rekey-limit time** *minutes*

Example:

```
Firepower /system/services # set /system/services # set ssh-client rekey-limit time ?
   10-1440  Max time limit in Minutes
```

Step 18 Commit the transaction to the system configuration:

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

Example

The following example enables SSH access to the chassis and commits the transaction:

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

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

This section describes how 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 Firepower 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.

Model	Level	Authentication	Encryption	What Happens
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 Enter monitoring mode:

Firepower-chassis# **scope monitoring**

Step 2 Enable SNMP:

Firepower-chassis /monitoring # **enable snmp**

Step 3 (Optional) Enter SNMP community mode:

Firepower-chassis /monitoring # **set snmp community**

After you enter the **set snmp community** command, you are prompted to enter the SNMP community name.

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.

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.

Step 4 Specify the SNMP community name; this community name is used as a SNMP password. The community name can be any alphanumeric string up to 32 characters.

Firepower-chassis /monitoring # **Enter a snmp community:** *community-name*

There can be only one community name; however, you can use **set snmp community** to overwrite the existing name. To delete an existing community name (also disabling SNMP versions 1 and 2c for polling requests from the SNMP remote manager), enter **set snmp community** but do not type a community string; that is, simply press **`Enter`** again. After you commit the buffer, **show snmp** output will include the line `Is Community Set: No`.

Step 5 Specify the system contact person responsible for SNMP. The system contact name can be any alphanumeric string up to 255 characters, such as an email address or name and telephone number.

Firepower-chassis /monitoring # **set snmp syscontact** *system-contact-name*

Step 6 Specify the location of the host on which the SNMP agent (server) runs. The system location name can be any alphanumeric string up to 512 characters.

Firepower-chassis /monitoring # **set snmp syslocation** *system-location-name*

Step 7 Commit the transaction to the system configuration:

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

Example

The following example enables SNMP, configures an SNMP community named SnmpCommSystem2, configures a system contact named contactperson, configures a contact location named systemlocation, and commits the transaction:

```
Firepower-chassis# scope ssa
Firepower-chassis# show app-instance
App Name   Identifier Slot ID    Admin State Oper State       Running Version Startup Version
 Deploy Type Turbo Mode Profile Name Cluster State   Cluster Role
---------- ---------- ---------- ----------- ---------------- --------------- ---------------
 ----------- ---------- ------------ --------------- ------------
ftd        ftd1       1          Enabled     Online           7.2.0.82        7.2.0.82
    Native      No                      Not Applicable  None
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # enable snmp
Firepower-chassis /monitoring # set snmp adminappinstance slot 1 appname ftd id ftd1 enable
 yes
Firepower-chassis /monitoring* # set snmp community
Enter a snmp community: SnmpCommSystem2
Firepower-chassis /monitoring* # set snmp syscontact contactperson1
Firepower-chassis /monitoring* # set snmp syslocation systemlocation
Firepower-chassis /monitoring* # commit-buffer
Firepower-chassis /monitoring #
```

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 Enter monitoring mode:

Firepower-chassis# **scope monitoring**

Step 2 Enable SNMP:

Firepower-chassis /monitoring # **enable snmp**

Step 3 Create an SNMP trap with the specified host name, IPv4 address, or IPv6 address.

Firepower-chassis /monitoring # **create snmp-trap** {*hostname* | *ip-addr* | *ip6-addr*}

Step 4 Specify the SNMP community string, or version 3 user name, to be used with the SNMP trap:

Firepower-chassis /monitoring/snmp-trap # **set community** *community-name*

Specifies the SNMPv1/v2c community string, or the SNMPv3 user name, to permit access to the trap destination. You are queried for the community name after you enter this command. The name can be up to 32 characters with no spaces; the name is not displayed as you type.

Step 5 Specify the port to be used for the SNMP trap:

Firepower-chassis /monitoring/snmp-trap # **set port** *port-num*

Step 6 Specify the SNMP version and model used for the trap:

Firepower-chassis /monitoring/snmp-trap # **set version** {**v1** | **v2c** | **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.

Step 7 (Optional) Specify the type of trap to send.

Firepower-chassis /monitoring/snmp-trap # **set notificationtype** {**traps** | **informs**}

This can be:

- **traps** if you select v2c or v3 for the version.
- **informs** if you select v2c for the version.

  Note An inform notification can be sent only if you select v2c for the version.

Step 8 (Optional) If you select v3 for the version, specify the privilege associated with the trap:

Firepower-chassis /monitoring/snmp-trap # **set v3privilege** {**auth** | **noauth** | **priv**}

This can be:

- **auth**—Authentication but no encryption.
- **noauth**—No authentication or encryption. Note that while you can specify it, FXOS does not support this security level with SNMPv3.
- **priv**—Authentication and encryption.

Step 9 Commit the transaction to the system configuration:

Firepower-chassis /monitoring/snmp-trap # **commit-buffer**

Example

The following example enables SNMP, creates an SNMP trap using an IPv4 address, specifies that the trap will use the SnmpCommSystem2 community on port 2, sets the version to v3, sets the notification type to traps, sets the v3 privilege to priv, and commits the transaction:

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # enable snmp
Firepower-chassis /monitoring* # create snmp-trap 192.168.100.112
Firepower-chassis /monitoring/snmp-trap* # set community SnmpCommSystem2
Firepower-chassis /monitoring/snmp-trap* # set port 2
Firepower-chassis /monitoring/snmp-trap* # set version v3
Firepower-chassis /monitoring/snmp-trap* # set notificationtype traps
Firepower-chassis /monitoring/snmp-trap* # set v3privilege priv
Firepower-chassis /monitoring/snmp-trap* # commit-buffer
Firepower-chassis /monitoring/snmp-trap #
```

The following example enables SNMP, creates an SNMP trap using an IPv6 address, specifies that the trap will use the SnmpCommSystem3 community on port 2, sets the version to v3, sets the notification type to traps, sets the v3 privilege to priv, and commits the transaction:

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # enable snmp
Firepower-chassis /monitoring* # create snmp-trap 2001::1
Firepower-chassis /monitoring/snmp-trap* # set community SnmpCommSystem3
Firepower-chassis /monitoring/snmp-trap* # set port 2
Firepower-chassis /monitoring/snmp-trap* # set version v3
Firepower-chassis /monitoring/snmp-trap* # set notificationtype traps
Firepower-chassis /monitoring/snmp-trap* # set v3privilege priv
Firepower-chassis /monitoring/snmp-trap* # commit-buffer
Firepower-chassis /monitoring/snmp-trap #
```

Deleting an SNMP Trap

Procedure

Step 1 Enter monitoring mode:

Firepower-chassis# **scope monitoring**

Step 2 Delete the SNMP trap with the specified hostname or IP address:

Firepower-chassis /monitoring # **delete snmp-trap** {*hostname* | *ip-addr*}

Step 3 Commit the transaction to the system configuration:

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

Example

The following example deletes the SNMP trap at IP address 192.168.100.112 and commits the transaction:

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # delete snmp-trap 192.168.100.112
Firepower-chassis /monitoring* # commit-buffer
Firepower-chassis /monitoring #
```

Creating an SNMPv3 User

Procedure

Step 1 Enter monitoring mode:

Firepower-chassis# **scope monitoring**

Step 2 Enable SNMP:

Firepower-chassis /monitoring # **enable snmp**

Step 3 Create an SNMPv3 user:

Firepower-chassis /monitoring # **create snmp-user** *user-name*

After you enter the **create snmp-user** command, you are prompted to enter a password.

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.

Step 4 Specify the use of SHA authentication:

Firepower-chassis /monitoring/snmp-user # **set auth** [**sha** | **sha224** | **sha256** | **sha358**}

Step 5 Enable or disable the use of AES-128 encryption:

Firepower-chassis /monitoring/snmp-user # **set aes-128** {**no** | **yes**}

By default, AES-128 encryption is disabled.

SNMPv3 does not support DES. If you leave AES-128 disabled, no privacy encryption will be done and any configured privacy password will have no effect.

Note You cannot poll SNMPv3 FXOS device from certain NMS monitoring applications when SNMPv3 with Authpriv (DES) is enabled. If you upgrade the device from a version that supported using DES previously, you must recreate the users using AES to poll the SNMPv3 FXOS device.

Step 6 Specify the user password:

Firepower-chassis /monitoring/snmp-user # **set password**

After you enter the **set password** command, you are prompted to enter and confirm the password.

Step 7 Commit the transaction to the system configuration:

Firepower-chassis /monitoring/snmp-user # **commit-buffer**

Example

The following example enables SNMP, creates an SNMPv3 user named snmp-user14, enables AES-128 encryption, sets the password and privacy password, and commits the transaction:

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # enable snmp
Firepower-chassis /monitoring* # create snmp-user snmp-user14
Password:
Firepower-chassis /monitoring/snmp-user* # set aes-128 yes
Firepower-chassis /monitoring/snmp-user* # set priv-password
Enter a password:
Confirm the password:
Firepower-chassis /monitoring/snmp-user* # commit-buffer
Firepower-chassis /monitoring/snmp-user #
```

Deleting an SNMPv3 User

Procedure

Step 1 Enter monitoring mode:

Firepower-chassis# **scope monitoring**

Step 2 Delete the specified SNMPv3 user:

Firepower-chassis /monitoring # **delete snmp-user** *user-name*

Step 3 Commit the transaction to the system configuration:

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

Example

The following example deletes the SNMPv3 user named snmp-user14 and commits the transaction:

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # delete snmp-user snmp-user14
Firepower-chassis /monitoring* # commit-buffer
Firepower-chassis /monitoring #
```

Viewing Current SNMP Settings

Use the following CLI commands to display current SNMP settings, users and traps.

Note The ifIndex order on the interface of FXOS that uses SNMP does not change after you reboot the FXOS.

Procedure

Step 1 Enter monitoring mode:

firepower# **scope monitoring**

Step 2 Display the current SNMP settings:

firepower/monitoring # **show snmp**

```
Name: snmp
    Admin State: Enabled
    Port: 161
    Is Community Set: Yes
    Sys Contact: R_Admin
    Sys Location:
```

Step 3 List the currently defined SNMPv3 users:

firepower/monitoring # **show snmp-user**

```
SNMPv3 User:
    Name                     Authentication type
    ------------------------ -------------------
    snmp-user1               Sha
    testuser                 Sha
    snmp-user2               Sha
```

Step 4 List the currently defined SNMP traps:

firepower/monitoring # **show snmp-trap**

```
SNMP Trap:
    SNMP Trap                Port     Community  Version V3 Privilege Notification Type
    ------------------------ -------- ---------- ------- ------------ -----------------
    trap1_informs            162      ****       V2c     Noauth       Informs
    192.168.10.100           162      ****       V3      Noauth       Traps
```

Example

This example show how to display detailed information about a specific SNMPv3 user:

```
firepower /monitoring # show snmp-user snmp-user1 detail

SNMPv3 User:
    Name: snmp-user1
    Authentication type: Sha
    Password: ****
    Privacy password: ****
    Use AES-128: Yes
firepower /monitoring #
```

Configuring HTTPS

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

Note You can change the HTTPS port using Firepower 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 36.

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 (Optional) Enable or disable the certificate revocation list check:

set revoke-policy *{ relaxed | strict }*

Step 9 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 Enter system mode:

Firepower-chassis # **scope system**

Step 2 Enter system services mode:

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

Step 3 Specify the port to use for HTTPS connections:

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

Specify an integer between 1 and 65535 for *port-number*. HTTPS is enabled on port 443 by default.

Step 4 Commit the transaction to the system configuration:

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

After changing the HTTPS port, all current HTTPS sessions are closed. Users will need to log back in to the Firepower 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.

Example

The following example sets the HTTPS port number to 443 and commits the transaction:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # set https port 444
Warning: When committed, this closes all the web sessions.
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

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+
- **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 40
     - Configuring RADIUS Providers, on page 45
     - Configuring TACACS+ Providers, on page 48

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 Enter security mode:

Firepower-chassis# **scope security**

Step 2 Enter security LDAP mode:

Firepower-chassis /security # **scope ldap**

Step 3 Restrict database searches to records that contain the specified attribute:

Firepower-chassis /security/ldap # **set attribute** *attribute*

Step 4 Restrict database searches to records that contain the specified distinguished name:

Firepower-chassis /security/ldap # **set basedn** *distinguished-name*

Step 5 Restrict database searches to records that contain the specified filter:

Firepower-chassis /security/ldap # **set filter** *filter*

where *filter* is the filter attribute to use with your LDAP server, for example *cn=$userid* or *sAMAccountName=$userid*. The filter must include *$userid*.

Step 6 Set the amount of time the system will wait for a response from the LDAP server before noting the server as down:

Firepower-chassis /security/ldap # **set timeout** *seconds*

Step 7 Commit the transaction to the system configuration:

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

Example

The following example sets the LDAP attribute to CiscoAvPair, the base distinguished name to "DC=cisco-firepower-aaa3,DC=qalab,DC=com", the filter to sAMAccountName=$userid, and the timeout interval to 5 seconds, and commits the transaction:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope ldap
Firepower-chassis /security/ldap # set attribute CiscoAvPair
Firepower-chassis /security/ldap* # set basedn "DC=cisco-firepower-aaa3,DC=qalab,DC=com"
Firepower-chassis /security/ldap* # set filter sAMAccountName=$userid
Firepower-chassis /security/ldap* # set timeout 5
Firepower-chassis /security/ldap* # commit-buffer
Firepower-chassis /security/ldap #
```

Note User login will fail if the DN for an LDAP user exceeds 255 characters.

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 Enter security mode:

Firepower-chassis# **scope security**

Step 2 Enter security LDAP mode:

Firepower-chassis /security # **scope ldap**

Step 3 Create an LDAP server instance and enter security LDAP server mode:

Firepower-chassis /security/ldap # **create server** *server-name*

If SSL is enabled, the *server-name* , typically an IP address or FQDN, must exactly match a Common Name (CN) in the LDAP server's security certificate. Unless an IP address is specified, a DNS server must be configured.

Step 4 (Optional) Set an LDAP attribute that stores the values for the user roles and locales:

Firepower-chassis /security/ldap/server # **set attribute** *attr-name*

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 for LDAP providers.

Step 5 (Optional) Set 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:

Firepower-chassis /security/ldap/server # **set basedn** *basedn-name*

The length of the base DN can be a maximum of 255 characters minus the length of CN=username, where username identifies the remote user attempting to access Firepower Chassis Manager or the FXOS CLI using LDAP authentication.

This value is required unless a default base DN has been set for LDAP providers.

Step 6 (Optional) Set the distinguished name (DN) for an LDAP database account that has read and search permissions for all objects under the base DN:

Firepower-chassis /security/ldap/server # **set binddn** *binddn-name*

The maximum supported string length is 255 ASCII characters.

Step 7 (Optional) Restrict the LDAP search to user names that match the defined filter.

Firepower-chassis /security/ldap/server # **set filter** *filter-value*

where *filter-value* is the filter attribute to use with your LDAP server; for example *cn=$userid* or *sAMAccountName=$userid*. The filter must include *$userid*.

This value is required unless a default filter has been set for LDAP providers.

Step 8 Specify the password for the LDAP database account specified for Bind DN:

Firepower-chassis /security/ldap/server # **set password**

To set the password, press **Enter** after typing the **set password** command and enter the key value at the prompt.

You can enter any standard ASCII characters except for space, § (section sign), ? (question mark), or = (equal sign).

Step 9 (Optional) Specify the order in which the FXOS uses this provider to authenticate users:

Firepower-chassis /security/ldap/server # **set order** *order-num*

Step 10 (Optional) Specify the port used to communicate with the LDAP server. The standard port number is 389.

Firepower-chassis /security/ldap/server # **set port** *port-num*

Step 11 Enable or disable the use of encryption when communicating with the LDAP server:

Firepower-chassis /security/ldap/server # **set ssl** {**yes** | **no**}

The options are as follows:

- **yes** —Encryption is required. If encryption cannot be negotiated, the connection fails.
- **no** —Encryption is disabled. Authentication information is sent as clear text.

LDAP uses STARTTLS. This allows encrypted communication using port 389.

Step 12 Specify the length of time in seconds the system will spend trying to contact the LDAP database before it times out:

Firepower-chassis /security/ldap/server # **set timeout** *timeout-num*

Enter an integer from 1 to 60 seconds, or enter 0 (zero) to use the global timeout value specified for LDAP providers. The default is 30 seconds.

Step 13 Specify the vendor that is providing the LDAP provider or server details:

Firepower-chassis /security/ldap/server # **set vendor** *{ms-ad* | *openldap}*

The options are as follows:

- **ms-ad**—LDAP provider is Microsoft Active Directory.
- **openldap**—LDAP provider is not Microsoft Active Directory.

Step 14 (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.

Step 15 Commit the transaction to the system configuration:

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

Example

The following example creates an LDAP server instance named 10.193.169.246, configures the binddn, password, order, port, SSL settings, vendor attribute, and commits the transaction:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope ldap
Firepower-chassis /security/ldap* # create server 10.193.169.246
Firepower-chassis /security/ldap/server* # set binddn
"cn=Administrator,cn=Users,DC=cisco-firepower-aaa3,DC=qalab,DC=com"
Firepower-chassis /security/ldap/server* # set password
Enter the password:
Confirm the password:
Firepower-chassis /security/ldap/server* # set order 2
Firepower-chassis /security/ldap/server* # set port 389
Firepower-chassis /security/ldap/server* # set ssl yes
Firepower-chassis /security/ldap/server* # set timeout 30
Firepower-chassis /security/ldap/server* # set vendor ms-ad
Firepower-chassis /security/ldap/server* # commit-buffer
Firepower-chassis /security/ldap/server #
```

The following example creates an LDAP server instance named 12:31:71:1231:45b1:0011:011:900, configures the binddn, password, order, port, SSL settings, vendor attribute, and commits the transaction:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope ldap
Firepower-chassis /security/ldap* # create server 12:31:71:1231:45b1:0011:011:900
Firepower-chassis /security/ldap/server* # set binddn
"cn=Administrator,cn=Users,DC=cisco-firepower-aaa3,DC=qalab,DC=com"
Firepower-chassis /security/ldap/server* # set password
Enter the password:
Confirm the password:
Firepower-chassis /security/ldap/server* # set order 1
Firepower-chassis /security/ldap/server* # set port 389
Firepower-chassis /security/ldap/server* # set ssl yes
Firepower-chassis /security/ldap/server* # set timeout 45
Firepower-chassis /security/ldap/server* # set vendor ms-ad
Firepower-chassis /security/ldap/server* # commit-buffer
Firepower-chassis /security/ldap/server #
```

Deleting an LDAP Provider

Procedure

Step 1 Enter security mode:

Firepower-chassis# **scope security**

Step 2 Enter security LDAP mode:

Firepower-chassis /security # **scope ldap**

Step 3 Delete the specified server:

Firepower-chassis /security/ldap # **delete server** *serv-name*

Step 4 Commit the transaction to the system configuration:

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

Example

The following example deletes the LDAP server called ldap1 and commits the transaction:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope ldap
Firepower-chassis /security/ldap # delete server ldap1
Firepower-chassis /security/ldap* # commit-buffer
Firepower-chassis /security/ldap #
```

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 Enter security mode:

Firepower-chassis# **scope security**

Step 2 Enter security RADIUS mode:

Firepower-chassis /security # **scope radius**

Step 3 (Optional) Specify the number of times to retry contacting the RADIUS server before noting the server as down:

Firepower-chassis /security/radius # **set retries** *retry-num*

Step 4 (Optional) Set the amount of time the system will wait for a response from the RADIUS server before noting the server as down:

Firepower-chassis /security/radius # **set timeout** *seconds*

Step 5 Commit the transaction to the system configuration:

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

Example

The following example sets the RADIUS retries to 4, sets the timeout interval to 30 seconds, and commits the transaction:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope radius
Firepower-chassis /security/radius # set retries 4
Firepower-chassis /security/radius* # set timeout 30
Firepower-chassis /security/radius* # commit-buffer
Firepower-chassis /security/radius #
```

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 Enter security mode:

Firepower-chassis# **scope security**

Step 2 Enter security RADIUS mode:

Firepower-chassis /security # **scope radius**

Step 3 Create a RADIUS server instance and enter security RADIUS server mode:

Firepower-chassis /security/radius # **create server** *server-name*

Step 4 (Optional) Specify the port used to communicate with the RADIUS server.

Firepower-chassis /security/radius/server # **set authport** *authport-num*

Step 5 Set the RADIUS server key:

Firepower-chassis /security/radius/server # **set key**

To set the key value, press **Enter** after typing the **set key** command and enter the key value at the prompt.

You can enter any standard ASCII characters except for space, § (section sign), ? (question mark), or = (equal sign).

Step 6 (Optional) Specify when in the order this server will be tried:

Firepower-chassis /security/radius/server # **set order** *order-num*

Step 7 (Optional) Set the number of times to retry communicating with the RADIUS server before noting the server as down:

Firepower-chassis /security/radius/server # **set retries** *retry-num*

Step 8 Specify the length of time in seconds the system will wait for a response from the RADIUS server before noting the server as down:

Firepower-chassis /security/radius/server # **set timeout** *seconds*

Tip It is recommended that you configure a higher **Timeout** value if you select two-factor authentication for RADIUS providers.

Step 9 Commit the transaction to the system configuration:

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

Example

The following example creates a server instance named radiusserv7, sets the authentication port to 5858, sets the key to radiuskey321, sets the order to 2, sets the retries to 4, sets the timeout to 30, and commits the transaction:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope radius
Firepower-chassis /security/radius # create server radiusserv7
Firepower-chassis /security/radius/server* # set authport 5858
Firepower-chassis /security/radius/server* # set key
Enter the key: radiuskey321
Confirm the key: radiuskey321
Firepower-chassis /security/radius/server* # set order 2
Firepower-chassis /security/radius/server* # set retries 4
Firepower-chassis /security/radius/server* # set timeout 30
Firepower-chassis /security/radius/server* # commit-buffer
Firepower-chassis /security/radius/server #
```

Deleting a RADIUS Provider

Procedure

Step 1 Enter security mode:

Firepower-chassis# **scope security**

Step 2 Enter security RADIUS mode:

Firepower-chassis /security # **scope RADIUS**

Step 3 Delete the specified server:

Firepower-chassis /security/radius # **delete server** *serv-name*

Step 4 Commit the transaction to the system configuration:

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

Example

The following example deletes the RADIUS server called radius1 and commits the transaction:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope radius
Firepower-chassis /security/radius # delete server radius1
Firepower-chassis /security/radius* # commit-buffer
Firepower-chassis /security/radius #
```

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 Enter security mode:

Firepower-chassis# **scope security**

Step 2 Enter security TACACS+ mode:

Firepower-chassis /security # **scope tacacs**

Step 3 (Optional) Set the amount of time the system will wait for a response from the TACACS+ server before noting the server as down:

Firepower-chassis /security/tacacs # **set timeout** *seconds*

Enter an integer from 1 to 60 seconds. The default value is 5 seconds.

Step 4 Commit the transaction to the system configuration:

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

Example

The following example sets the TACACS+ timeout interval to 45 seconds and commits the transaction:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope tacacs
Firepower-chassis /security/tacacs # set timeout 45
```

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

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 Enter security mode:

Firepower-chassis# **scope security**

Step 2 Enter security TACACS+ mode:

Firepower-chassis /security # **scope tacacs**

Step 3 Create a TACACS+ server instance and enter security TACACS+ server mode:

Firepower-chassis /security/tacacs # **create server** *server-name*

Step 4 Specify the TACACS+ server key:

Firepower-chassis /security/tacacs/server # **set key**

To set the key value, press **Enter** after typing the **set key** command and enter the key value at the prompt.

You can enter any standard ASCII characters except for space, § (section sign), ? (question mark), or = (equal sign).

Step 5 (Optional) Specify when in the order this server will be tried:

Firepower-chassis /security/tacacs/server # **set order** *order-num*

Step 6 Specify the time interval that the system will wait for a response from the TACACS+ server before noting the server as down:

Firepower-chassis /security/tacacs/server # **set timeout** *seconds*

Tip It is recommended that you configure a higher timeout value if you select two-factor authentication for TACACS+ providers.

Step 7 (Optional) Specify the port used to communicate with the TACACS+ server:

Firepower-chassis /security/tacacs/server # **set port** *port-num*

Step 8 Commit the transaction to the system configuration:

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

Example

The following example creates a server instance named tacacsserv680, sets the key to tacacskey321, sets the order to 4, sets the authentication port to 5859, and commits the transaction:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope tacacs
Firepower-chassis /security/tacacs # create server tacacsserv680
Firepower-chassis /security/tacacs/server* # set key
Enter the key: tacacskey321
Confirm the key: tacacskey321
Firepower-chassis /security/tacacs/server* # set order 4
Firepower-chassis /security/tacacs/server* # set port 5859
Firepower-chassis /security/tacacs/server* # commit-buffer
Firepower-chassis /security/tacacs/server #
```

Deleting a TACACS+ Provider

Procedure

Step 1 Enter security mode:

Firepower-chassis# **scope security**

Step 2 Enter security TACACS+ mode:

Firepower-chassis /security # **scope tacacs**

Step 3 Delete the specified server:

Firepower-chassis /security/tacacs # **delete server** *serv-name*

Step 4 Commit the transaction to the system configuration:

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

Example

The following example deletes the TACACS+ server called tacacs1 and commits the transaction:

```
Firepower-chassis# scope security
Firepower-chassis /security # scope tacacs
Firepower-chassis /security/tacacs # delete server tacacs1
Firepower-chassis /security/tacacs* # commit-buffer
Firepower-chassis /security/tacacs #
```

Verifying Remote AAA Server Configurations

The following sections describe how to use the FXOS CLI to determine the current configuration for the various remote AAA servers.

Determining Current FXOS Authentication Configuration

The following example shows you how to use the **show authentication** command to determine the current FXOS authentication settings. In this example, LDAP is the default mode of authentication.

```
firepower# scope security
firepower /security # show authentication
Console authentication: Local
Operational Console authentication: Local
Default authentication: Ldap
Operational Default authentication: Ldap
Role Policy For Remote Users: Assign Default Role
firepower /security #
```

Determining Current LDAP Configuration

The following example shows you how to use the **show server detail** command in ldap mode to determine the current LDAP configuration settings.

```
firepower# scope security
firepower /security # scope ldap
firepower /security/ldap # show server detail

LDAP server:
    Hostname, FQDN or IP address: 10.48.53.132
    Descr:
    Order: 1
    DN to search and read: CN=cisco,CN=Users,DC=fxosldapuser,DC=lab
    Password:
    Port: 389
    SSL: No
    Key:
    Cipher Suite Mode: Medium Strength
    Cipher Suite:
ALL:!DHE-PSK-AES256-CBC-SHA:!EDH-RSA-DES-CBC3-SHA:!EDH-DSS-DES-CBC3-SHA:!DES-CBC3-SHA:!ADH:!3DES:!EXPORT40:!EXPORT56:!LOW:!RC4:!MD5:!IDEA:+HIGH:+MEDIUM:+EXP:+eNULL

    CRL: Relaxed
    Basedn: CN=Users,DC=fxosldapuser,DC=lab
    User profile attribute: CiscoAVPair
    Filter: cn=$userid
    Timeout: 30
    Ldap Vendor: MS AD
firepower /security/ldap #
```

Determining Current RADIUS Configuration

The following example shows you how to use the **show server detail** command in radius mode to determine the current RADIUS configuration settings.

```
firepower# scope security
firepower /security # scope radius
firepower /security/radius # show server detail
```

```
RADIUS server:
    Hostname, FQDN or IP address: 10.48.17.199
    Descr:
    Order: 1
    Auth Port: 1812
    Key: ****
    Timeout: 5
    Retries: 1
firepower /security/radius #
```

Determining Current TACACS+ Configuration

The following example shows you how to use the **show server detail** command in tacacs mode to determine the current TACACS+ configuration settings.

```
firepower# scope security
firepower /security # scope tacacs
firepower /security/tacacs # show server detail

TACACS+ server:
    Hostname, FQDN or IP address: 10.48.17.199
    Descr:
    Order: 1
    Port: 49
    Key: ****
    Timeout: 5
firepower /security/tacacs #
```

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 Enter monitoring mode:

Firepower-chassis# **scope monitoring**

Step 2 Enable or disable the sending of syslogs to the console:

Firepower-chassis /monitoring # {**enable** | **disable**} **syslog console**

Step 3 (Optional) Select the lowest message level that you want displayed. If syslogs are enabled, the system displays that level and above on the console. The level options are listed in order of decreasing urgency. The default level is Critical.

Firepower-chassis /monitoring # **set syslog console level** {**emergencies** | **alerts** | **critical**}

Step 4 Enable or disable the monitoring of syslog information by the operating system:

Firepower-chassis /monitoring # {**enable** | **disable**} **syslog monitor**

Step 5 (Optional) Select the lowest message level that you want displayed. If the monitor state is enabled, the system displays that level and above. The level options are listed in order of decreasing urgency. The default level is Critical.

Firepower-chassis /monitoring # **set syslog monitor level** {**emergencies** | **alerts** | **critical** | **errors** | **warnings** | **notifications** | **information** | **debugging**}

Note Messages at levels below Critical are displayed on the terminal monitor only if you have entered the **terminal monitor** command.

Step 6 Enable or disable the writing of syslog information to a syslog file:

Firepower-chassis /monitoring # {**enable** | **disable**} **syslog file**

Step 7 Specify the name of the file in which the messages are logged. Up to 16 characters are allowed in the file name.

Firepower-chassis /monitoring # **set syslog file name** *filename*

Step 8 (Optional) Select the lowest message level that you want stored to a file. If the file state is enabled, the system stores that level and above in the syslog file. The level options are listed in order of decreasing urgency. The default level is Critical.

Firepower-chassis /monitoring # **set syslog file level** {**emergencies** | **alerts** | **critical** | **errors** | **warnings** | **notifications** | **information** | **debugging**}

Step 9 (Optional) Specify the maximum file size, in bytes, before the system begins to write over the oldest messages with the newest ones. The range is 4096 to 4194304 bytes.

Firepower-chassis /monitoring # **set syslog file size** *filesize*

Step 10 Configure sending of syslog messages to up to three external syslog servers:

a) Enable or disable the sending of syslog messages to up to three external syslog servers:

Firepower-chassis /monitoring # {**enable** | **disable**} **syslog remote-destination** {**server-1** | **server-2** | **server-3**}

b) (Optional) Select the lowest message level that you want stored to the external log. If the remote-destination is enabled, the system sends that level and above to the external server. The level options are listed in order of decreasing urgency. The default level is Critical.

Firepower-chassis /monitoring # **set syslog remote-destination** {**server-1** | **server-2** | **server-3**} **level**{**emergencies** | **alerts** | **critical** | **errors** | **warnings** | **notifications** | **information** | **debugging**}

c) Specify the hostname or IP address of the specified remote syslog server. Up to 256 characters are allowed in the hostname.

Firepower-chassis /monitoring # **set syslog remote-destination** {**server-1** | **server-2** | **server-3**} **hostname** *hostname*

d) (Optional) Specify the facility level contained in the syslog messages sent to the specified remote syslog server.

Firepower-chassis /monitoring # **set syslog remote-destination** {**server-1** | **server-2** | **server-3**} **facility** {**local0** | **local1** | **local2** | **local3** | **local4** | **local5** | **local6** | **local7**}

Step 11 Configure the local sources. Enter the following command for each of the local sources you want to enable or disable:

Firepower-chassis /monitoring # {**enable** | **disable**} **syslog source** {**audits** | **events** | **faults**}

This can be one of the following:

- **audits**—Enables or disables the logging of all audit log events.
- **events**—Enables or disables the logging of all system events.
- **faults**—Enables or disables the logging of all system faults.

Step 12 Commit the transaction:

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

Example

This example shows how to enable the storage of syslog messages in a local file and commits the transaction:

```
Firepower-chassis# scope monitoring
Firepower-chassis /monitoring # disable syslog console
Firepower-chassis /monitoring* # disable syslog monitor
Firepower-chassis /monitoring* # enable syslog file
Firepower-chassis /monitoring* # set syslog file name SysMsgsFirepower
Firepower-chassis /monitoring* # set syslog file level notifications
Firepower-chassis /monitoring* # set syslog file size 4194304
Firepower-chassis /monitoring* # disable syslog remote-destination server-1
Firepower-chassis /monitoring* # disable syslog remote-destination server-2
Firepower-chassis /monitoring* # disable syslog remote-destination server-3
Firepower-chassis /monitoring* # commit-buffer
Firepower-chassis /monitoring #
```

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 Enter system mode:

Firepower-chassis # **scope system**

Step 2 Enter system services mode:

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

Step 3 To create or delete a DNS server, enter the appropriate command as follows:

- To configure the system to use a DNS server with the specified IPv4 or IPv6 address:

  Firepower-chassis /system/services # **create dns** *{ip-addr* | *ip6-addr}*

- To delete a DNS server with the specified IPv4 or IPv6 address:

  Firepower-chassis /system/services # **delete dns** *{ip-addr* | *ip6-addr}*

Step 4 Commit the transaction to the system configuration:

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

Example

The following example configures a DNS server with the IPv4 address 192.168.200.105 and commits the transaction:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # create dns 192.168.200.105
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

The following example configures a DNS server with the IPv6 address 2001:db8::22:F376:FF3B:AB3F and commits the transaction:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # create dns 2001:db8::22:F376:FF3B:AB3F
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

The following example deletes the DNS server with the IP address 192.168.200.105 and commits the transaction:

```
Firepower-chassis# scope system
Firepower-chassis /system # scope services
Firepower-chassis /system/services # delete dns 192.168.200.105
Firepower-chassis /system/services* # commit-buffer
Firepower-chassis /system/services #
```

Enable FIPS Mode

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

Procedure

Step 1 From the FXOS CLI, enter the security mode:

scope security

Step 2 Enable FIPS mode:

enable fips-mode

Step 3 Commit the configuration:

commit-buffer

Step 4 Reboot the system:

connect local-mgmt

reboot

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 From the FXOS CLI, enter the security mode:

scope security

Step 2 Enable Common Criteria mode:

enable cc-mode

Step 3 Commit the configuration:

commit-buffer

Step 4 Reboot the system:

connect local-mgmt

reboot

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 From the FXOS CLI, enter the services mode:

scope system

scope services

Step 2 Create an IP block for the services you want to enable access for:

For IPv4:

create ip-block *ip prefix [0-32]* [**http** | **snmp** | **ssh**]

For IPv6:

create ipv6-block *ip prefix [0-128]* [**http** | **snmp** | **ssh**]

Example

The following example shows how to create, enter, and verify an IPv4 address block to provide SSH access:

```
firepower # scope system
firepower /system # scope services
firepower /system/services # enter ip-block 192.168.200.101 32 ssh
firepower /system/services/ip-block* # commit-buffer
firepower /system/services/ip-block # up
firepower /system/services # show ip-block

Permitted IP Block:
    IP Address      Prefix Length Protocol
    --------------- ------------- --------
    0.0.0.0                     0 https
    0.0.0.0                     0 snmp
    0.0.0.0                     0 ssh
    192.168.200.101            32 ssh
firepower /system/services #
```

The following example shows how to create, enter and verify an IPv6 address block to provide SSH access::

```
firepower # scope system
firepower /system # scope services
firepower /system/services # create ipv6-block 2001:DB8:1::1 64 ssh
firepower /system/services/ipv6-block* # commit-buffer
firepower /system/services/ipv6-block # up
firepower /system/services # show ipv6-block

Permitted IPv6 Block:
    IPv6 Address Prefix Length Protocol
    ------------ ------------- --------
    ::                       0 https
    ::                       0 snmp
    ::                       0 ssh
    2001:DB8:1::1
                            64 ssh
firepower /system/services #
```

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 Enter Security Services mode, and then Auto MAC pool mode.

scope ssa

scope auto-macpool

Example:

```
Firepower# scope ssa
Firepower /ssa # scope auto-macpool
Firepower /ssa/auto-macpool #
```

Step 2 Set the MAC address prefix used in generating the MAC addresses.

set prefix *prefix*

- *prefix*—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*

Example:

```
Firepower /ssa/auto-macpool # set prefix 65
Firepower /ssa/auto-macpool* #
```

Step 3 Save the configuration.

commit-buffer

Example:

```
Firepower /ssa/auto-macpool* # commit-buffer
Firepower /ssa/auto-macpool #
```

Step 4 View MAC address assignments.

show mac-address

Example:

```
Firepower /ssa/auto-macpool # show mac-address
Mac Address Item:
    Mac Address          Owner Profile            Owner Name
    -------------------- ------------------------ ----------
    A2:46:C4:00:00:1E    ftd13                    Port-channel14
    A2:46:C4:00:00:20    ftd14                    Port-channel15
    A2:46:C4:00:01:7B    ftd1                     Ethernet1/3
    A2:46:C4:00:01:7C    ftd12                    Port-channel11
    A2:46:C4:00:01:7D    ftd13                    Port-channel14
    A2:46:C4:00:01:7E    ftd14                    Port-channel15
    A2:46:C4:00:01:7F    ftd1                     Ethernet1/2
    A2:46:C4:00:01:80    ftd12                    Ethernet1/2
    A2:46:C4:00:01:81    ftd13                    Ethernet1/2
    A2:46:C4:00:01:82    ftd14                    Ethernet1/2
    A2:46:C4:00:01:83    ftd2                     Ethernet3/1/4
    A2:46:C4:00:01:84    ftd2                     Ethernet3/1/1
    A2:46:C4:00:01:85    ftd2                     Ethernet3/1/3
    A2:46:C4:00:01:86    ftd2                     Ethernet3/1/2
    A2:46:C4:00:01:87    ftd2                     Ethernet1/2
    A2:46:C4:00:01:88    ftd1                     Port-channel21
    A2:46:C4:00:01:89    ftd1                     Ethernet1/8
```

Example

The following example sets the MAC prefix to 33.

```
Firepower# scope ssa
Firepower /ssa # scope auto-macpool
Firepower /ssa/auto-macpool # set prefix 33
Firepower /ssa/auto-macpool* # commit-buffer
Firepower /ssa/auto-macpool #
```

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 Firepower Threat Defense instance to the FMC, then update the inventory for each unit on the FMC **Devices** > **Device Management** > **Device** > **System** > **Inventory** dialog box.

Procedure

Step 1 Enter Security Services mode.

scope ssa

Example:

```
Firepower# scope ssa
Firepower /ssa #
```

Step 2 Create the resource profile.

enter resource-profile *name*

- *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.

Example:

```
Firepower /ssa # enter resource-profile gold
Firepower /ssa/resource-profile* #
```

Step 3 Enter a description.

set description *description*

- *description*—Sets the description of the profile up to 510 characters. Use quotes (") around phrases with spaces.

Example:

```
Firepower /ssa/resource-profile* # set description "highest level"
```

Step 4 Set the number of CPU cores.

set cpu-core-count *cores*

- *cores*—Sets the number of cores for the profile, between 6 and the maximum, depending on your chassis, as an even number.

Example:

```
Firepower /ssa/resource-profile* # set cpu-core-count 14
```

Step 5 Save the configuration.

commit-buffer

Example:

```
Firepower /ssa/resource-profile* # commit-buffer
Firepower /ssa/resource-profile #
```

Step 6 View resource profile assignments from security services mode.

show resource-profile user-defined

Example:

```
Firepower /ssa # show resource-profile user-defined
Profile Name       Is In Use  CPU Logical Core Count Description
------------------ ---------- ---------------------- -----------
bronze               No                 6            low end device
gold                 No                 14           highest
silver               No                 10            mid-level
```

Step 7 View resource usage for the security module/engine slot.

show monitor detail

Example:

```
Firepower /ssa # scope slot 1
Firepower /ssa/slot # show monitor detail
Monitor:
    OS Version:
    CPU Total Load 1 min Avg: 18.959999
    CPU Total Load 5 min Avg: 19.080000
    CPU Total Load 15 min Avg: 19.059999
    Memory Total (MB): 252835
    Memory Free (MB): 200098
    Memory Used (MB): 52738
    CPU Cores Total: 72
    CPU Cores Available: 30
    Memory App Total (MB): 226897
    Memory App Available (MB): 97245
    Data Disk Total (MB): 1587858
    Data Disk Available (MB): 1391250
    Secondary Disk Total (MB): 0
    Secondary Disk Available (MB): 0
    Disk File System Count: 7
    Blade Uptime:
    Last Updated Timestamp: 2018-05-23T14:26:06.132
```

Step 8 View resource allocation for the application instance.

show resource detail

Example:

```
Firepower /ssa # scope slot 1
Firepower /ssa/slot # enter app-instance ftd ftd1
Firepower /ssa/slot/app-instance # show resource detail
Resource:
    Allocated Core NR: 10
    Allocated RAM (MB): 32413
    Allocated Data Disk (MB): 49152
    Allocated Binary Disk (MB): 3907
    Allocated Secondary Disk (MB): 0
```

Example

The following example adds three resource profiles.

```
Firepower# scope ssa
Firepower /ssa # enter resource-profile basic
Firepower /ssa/resource-profile* # set description "lowest level"
Firepower /ssa/resource-profile* # set cpu-core-count 6
Firepower /ssa/resource-profile* # exit
Firepower /ssa # enter resource-profile standard
Firepower /ssa/resource-profile* # set description "middle level"
Firepower /ssa/resource-profile* # set cpu-core-count 10
Firepower /ssa/resource-profile* # exit
Firepower /ssa # enter resource-profile advanced
Firepower /ssa/resource-profile* # set description "highest level"
Firepower /ssa/resource-profile* # set cpu-core-count 12
Firepower /ssa/resource-profile* # commit-buffer
Firepower /ssa/resource-profile #
```

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 Enter the organization scope.

scope org

Example:

```
Firepower # scope org
```

Step 2 Create and enable the network control policy.

create nw-ctrl-policy *nw-policy*

Example:

```
Firepower /org #  create nw-ctrl-policy nw-policy
```

Step 3 Enable LLDP.

enable lldp *{receive | transmit}*

Example:

```
Firepower /org/nw-ctrl-policy* # enable lldp receive
Firepower /org/nw-ctrl-policy* # disable lldp transmit
```

Step 4 Commit the configuration:

commit-buffer

Example:

```
Firepower /eth-uplink/fabric/interface* # commit-buffer
Firepower /eth-uplink/fabric/interface #
```

Step 5 Specify whether to enable or disable LLDP for receiving/transmitting.

enable lldp *receive/transmit*

commit-buffer

Example:

```
Firepower /org/nw-ctrl-policy* # enable lldp receive
Firepower /org/nw-ctrl-policy* # disable lldp transmit
Firepower /org/nw-ctrl-policy* # commit-buffer

Firepower /org/nw-ctrl-policy* # enable lldp receive
Firepower /org/nw-ctrl-policy* # disable lldp transmit
Firepower /org/nw-ctrl-policy* # commit-buffer
```

Step 6 Use the following commands to apply the netwok contol policy to an interface.

a) Enter the interface:

scope eth-uplink

scope fabric a

scope interface *interface_id*

```
Firepower # scope eth-uplink
Firepower /eth-uplink # scope fabric a
Firepower /eth-uplink/fabric # enter interface Ethernet3/1
```

b) Set the network control policy:

set nw-ctrl-policy *nw-policy*

commit-buffer

```
Firepower /eth-uplink/fabric/interface # set nw-ctrl-policy nw-policy
Firepower /eth-uplink/fabric/interface* # commit-buffer
MIO-5 /eth-uplink/fabric/interface # show detail
```

c) View the change:

show detail

```
Firepower /eth-uplink/fabric/interface # show detail
Interface:
    Port Name: Ethernet3/1
    User Label:
    Port Type: Data
    Admin State: Enabled
    Oper State: Sfp Not Present
    State Reason: Unknown
    flow control policy: default
    Auto negotiation: No
    Admin Speed: 100 Gbps
    Oper Speed: 100 Gbps
    Admin Duplex: Full Duplex
    Oper Duplex: Full Duplex
    Ethernet Link Profile name: default
    Oper Ethernet Link Profile name: fabric/lan/eth-link-prof-default
    Udld Oper State: Admin Disabled
    Inline Pair Admin State: Enabled
    Inline Pair Peer Port Name:
    Allowed Vlan: All
    Network Control Policy: nw-policy
    Current Task:
```

d) Commit the configuration:

commit-buffer

Example:

```
Firepower /eth-uplink/fabric/interface* # commit-buffer
Firepower /eth-uplink/fabric/interface #
```

Configure the Chassis URL

You can specify a management URL so that you can easily open Firepower Chassis Manager for an Firepower Threat Defense instance directly from FMC. 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 Firepower Threat Defense instance to the FMC, then update the inventory for each unit on the **Devices** > **Device Management** > **Device** > **System** > **Inventory** dialog box.

Procedure

Step 1 Enter the system mode:

scope system

Example:

```
Firepower# scope system
Firepower /system #
```

Step 2 To configure a new chassis name:

set name *chassis_name*

- *chassis_name*—Sets the name of the chassis between 1 and 60 characters.

Example:

```
Firepower /system # set name Firepower_chassis
```

Step 3 To configure the management URL:

set mgmt-url *management_url*

- *management_url*—Sets the URL that FMC should use to connect to an Firepower Threat Defense instance within Firepower Chassis Manager. The URL must start with https://. If you do not specify a chassis management URL, the chassis name is used instead.

Example:

```
Firepower /system # set mgmt-url https://192.168.1.55
```

Step 4 Save the configuration.

commit-buffer

Example:

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

Step 5 View configuration settings.

show detail

Example:

```
Firepower_chassis /system # show detail

Systems:
    Name: Firepower_chassis
    Mode: Stand Alone
    System IP Address: 192.168.1.10
    System IPv6 Address: ::
    System Owner:
    System Site:
    Description for System:
    Chassis Mgmt URL: https://192.168.1.55
```

Modifying Weak Key Exchange Algorithms

You can mitigate the weak key exchange algorithms used on the equipment by:

- Setting FIPS/CC Mode
- Setting Cipher Suite

Setting FIPS/CC Mode

Procedure

Step 1 From the FXOS CLI, enter the security mode:

scope security

Step 2 Enable FIPS mode:

enable fips-mode

Step 3 Commit the configuration:

commit-buffer

Setting Cipher Suite

Procedure

Step 1 Enter system mode:

Firepower-chassis# **scope system**

Step 2 Enter system services mode:

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

Step 3 View the HTTPS service:

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

Step 4 Set the Cipher Suite mode:

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

Step 5 Set the Cipher Suite string

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

Step 6 Commit the settings to the system configuration:

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