プラットフォーム設定

- 日時の設定 （1 ページ）
- Configuring SSH （8 ページ）
- TLS の設定 （13 ページ）
- Telnet の設定 （15 ページ）
- SNMP の設定 （16 ページ）
- HTTPS の設定 （27 ページ）
- AAA の設定 （41 ページ）
- リモート AAA サーバ設定の確認 （55 ページ）
- Syslog の設定 （57 ページ）
- DNS サーバの設定 （59 ページ）
- FIPS モードの有効化 （60 ページ）
- コモン クライテリア モードの有効化 （61 ページ）
- IP アクセスリストの設定 （62 ページ）
- MAC プール プレフィックスの追加とコンテナ インスタンス インターフェイスの MAC アドレスの表示 （64 ページ）
- コンテナインスタンスにリソースプロファイルを追加 （66 ページ）
- ネットワーク制御ポリシーの設定 （69 ページ）
- シャーシ URL の設定 （72 ページ）
- 脆弱キー交換アルゴリズムの変更 （73 ページ）

日時の設定

日付と時刻を手動で設定したり、現在のシステム時刻を表示するには、下記で説明するの CLI コマンドを使用してシステムのネットワーク タイム プロトコル（NTP）を設定します。

NTP の設定は、Firepower 4100/9300 シャーシとシャーシにインストールされている論理デバイス間で自動的に同期されます。

（注） Firepower 4100/9300 シャーシ に Firepower Threat Defense を導入すると、スマートライセンスが正しく機能し、デバイス登録に適切なタイムスタンプを確保するように Firepower 4100/9300 シャーシ で NTP を設定する必要があります。Firepower 4100/9300 シャーシ と FMC の両方で同じ NTP サーバーを使用する必要がありますが、FMC は Firepower 4100/9300 シャーシ の NTP サーバーとして使用できないので注意してください。

NTP を使用すると、[Current Time] タブの全体的な同期ステータスを表示できます。または、[Time Synchronization] タブの [NTP Server] テーブルの [Server Status] フィールドを見ると、設定済みの各 NTP サーバの同期ステータスを表示できます。システムが特定 NTP サーバと同期できない場合、[サーバのステータス（Server Status）] の横にある情報アイコンにカーソルを合わせると詳細を確認できます。

設定された日付と時刻の表示

手順

ステップ 1 FXOS CLI に接続します（FXOS CLIへのアクセスを参照）。

ステップ 2 設定されたタイム ゾーンを表示する場合：

Firepower-chassis# **show timezone**

ステップ 3 設定された日付と時刻を表示するには、次のコマンドを使用します。

Firepower-chassis# **show clock**

例

次の例では、設定されたタイム ゾーンと現在のシステム日時を表示する方法を示しています。

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

タイム ゾーンの設定

手順

ステップ 1 システム モードに入ります。

Firepower-chassis# **scope system**

ステップ 2 システム サービス モードを開始します。

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

ステップ 3 タイム ゾーンを設定します。

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

この時点で、大陸、国、およびタイム ゾーン領域に対応する番号を入力するように求められます。プロンプトごとに適切な情報を入力します。

ロケーション情報の指定を完了すると、プロンプトが表示され、正しいタイム ゾーン情報が設定されているか確認するよう求められます。確認する場合は**1**（yes）を入力し、操作をキャンセルする場合は **2**（no）を入力します。

ステップ 4 設定されたタイム ゾーンを表示するには：

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

Firepower-chassis# **show timezone**

例

次に、太平洋標準時領域にタイム ゾーンを設定し、トランザクションをコミットし、設定したタイム ゾーンを表示する例を示します。

```
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#
```

NTP を使用した日付と時刻の設定

NTP を使用して階層的なサーバ システムを実現し、ネットワーク システム間の時刻を正確に同期します。このような精度は、CRL の検証など正確なタイム スタンプを含む場合など、時刻が重要な操作で必要になります。 最大 4 台の NTP サーバを設定できます。

（注）
- FXOS では、NTP バージョン 3 を使用します。
- 外部 NTP サーバのストラタム値が 13 以上の場合、アプリケーションインスタンスは FXOS シャーシ上の NTP サーバと同期できません。NTP クライアントが NTP サーバと同期するたびに、ストラタム値が 1 ずつ増加します。

  独自の NTP サーバをセットアップしている場合は、サーバ上の /etc/ntp.conf ファイルでそのストラタム値を確認できます。NTP サーバのストラタム値が 13 以上の場合は、ntp.conf ファイルのストラタム値を変更してサーバを再起動するか、別の NTP サーバ（たとえば、pool.ntp.org）を使用することができます。

始める前に

NTP サーバのホスト名を使用する場合は、DNS サーバを設定する必要があります。DNS サーバの設定 （59 ページ）を参照してください。

手順

ステップ 1 システム モードに入ります。

Firepower-chassis# **scope system**

ステップ 2 システム サービス モードを開始します。

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

ステップ 3 指定したホスト名、IPv4 または IPv6 アドレスの NTP サーバを使用するようにシステムを設定します。

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

ステップ 4 （任意） NTP 認証を設定します。

NTP サーバ認証では SHA1 のみがサポートされます。NTP サーバからキー ID と値を取得します。たとえば、OpenSSL がインストールされた NTP サーバ バージョン 4.2.8 p8 以降で SHA1 キーを生成するには、**ntp-keygen -M** コマンドを入力して ntp. keys ファイルでキー ID と値を確認します。このキーは、クライアントとサーバの両方に対して、メッセージ ダイジェストの計算時に使用するキー値を通知するために使用します。

a) SHA1 キー ID を設定します。

set ntp-sha1-key-id *key_id*

b) SHA1 キー文字列を設定します。

set ntp-sha1-key-string

キー文字列を入力するように求められます。

c) ntp-server モードを修了します。

exit

d) NTP 認証をイネーブルにします。

enable ntp-authentication

例：

```
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
```

ステップ 5 トランザクションをシステム設定にコミットします。

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

ステップ 6 すべての設定済み NTP サーバの同期ステータスを表示するには、次のようにします。

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

ステップ 7 特定の NTP サーバの同期ステータスを表示するには、次のようにします。

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

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

例

次の例では、IP アドレス 192.168.200.101 を持つ NTP サーバを設定し、トランザクションをコミットします。

```
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 #
```

次に、IPv6 アドレス 4001::6 を持つ NTP サーバを設定し、トランザクションを確定する例を示します。

```
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 #
```

NTP サーバの削除

手順

ステップ 1 システム モードに入ります。

Firepower-chassis# **scope system**

ステップ 2 システム サービス モードを開始します。

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

ステップ 3 指定したホスト名、IPv4 または IPv6 アドレスの NTP サーバを削除します。

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

ステップ 4 トランザクションをシステム設定にコミットします。

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

例

次に、IP アドレス 192.168.200.101 の NTP サーバを削除し、トランザクションをコミットする例を示します。

```
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 #
```

次に、IPv6 アドレス 4001::6 を持つ NTP サーバを削除し、トランザクションを確定する例を示します。

```
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 #
```

日付と時刻の手動での設定

ここでは、シャーシで日付と時刻を手動で設定する方法ついて説明します。システムクロックの変更はシャーシでただちに有効になります。シャーシの日時を手動で設定した後、インス

トールされている論理デバイスに変更が反映されるまでに時間がかかる場合があることに注意してください。

（注） システム クロックが NTP サーバと同期中である場合は、日付と時刻を手動で設定することはできません。

手順

ステップ 1 システム モードに入ります。

Firepower-chassis# **scope system**

ステップ 2 システム サービス モードを開始します。

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

ステップ 3 システム クロックを設定します。

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

month には、月の英名の最初の 3 文字を使用します。時間は 24 時間形式で入力する必要があります。午後 7 時は 19 になります。

システム クロックの変更はただちに反映されます。バッファをコミットする必要はありません。

例

次に、システム クロックを設定する例を示します。

```
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

次の手順では、シャーシへの SSH アクセスを有効または無効にする方法、FXOS シャーシを SSH クライアントとして有効にする方法、さらに SSH で使用する暗号化、キー交換、およびメッセージ認証用のさまざまなアルゴリズムを SSH サーバーと SSH クライアントに設定する方法について説明します。

SSH はデフォルトでイネーブルになります。

手順

ステップ 1 システム モードに入ります。

Firepower-chassis # **scope system**

ステップ 2 システム サービス モードを開始します。

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

ステップ 3 シャーシへの SSH アクセスを設定するには、次のいずれかを実行します。

- シャーシへの SSH アクセスを許可するには、次のコマンドを入力します。

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

- シャーシへの SSH アクセスを禁止するには、次のコマンドを入力します。

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

ステップ 4 サーバの暗号化アルゴリズムを設定します。

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

例：

```
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
```

例：

（注）

- 次の暗号化アルゴリズムは、コモン クライテリア モードではサポートされていません。
  - 3des-cbc
  - chacha20-poly1305@openssh.com
- chacha20-poly1305@openssh.com は FIPS ではサポートされていません。FXOS シャーシで FIPS モードが有効になっている場合、chacha20-poly1305@openssh.com を暗号化アルゴリズムとして使用することはできません。
- 次の暗号化アルゴリズムは、デフォルトでは有効になっていません。

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

ステップ 5 サーバの Diffie-hellman（DH）キー交換アルゴリズムを設定します。

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

例：

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

DH キー交換では、いずれの当事者も単独では決定できない共有秘密を使用します。キー交換を署名およびホストキーと組み合わせることで、ホスト認証が実現します。このキー交換方式により、明示的なサーバ認証が可能となります。DH キー交換の使用方法の詳細については、RFC 4253 を参照してください。

（注）
- 次のキー交換アルゴリズムは、コモン クライテリア モードではサポートされていません。
  - diffie-hellman-group14-sha256
  - curve25519-sha256
  - curve25519-sha256@libssh.org
- FIPS モードでは、次のキー交換アルゴリズムはサポートされていません。
  - curve25519-sha256
  - curve25519-sha256@libssh.org

ステップ 6 サーバの MAC アルゴリズムを設定します。

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

例：

```
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
```

ステップ 7 サーバの ホスト キーについて、RSA キー ペアのモジュラス サイズを入力します。

モジュラス値（ビット単位）は、1024 ～ 2048 の範囲内の 8 の倍数です。指定するキー係数のサイズが大きいほど、RSA キー ペアの生成にかかる時間は長くなります。値は 2048 にすることをお勧めします。

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

例：

```
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
```

ステップ 8 サーバのキー再生成のボリューム制限について、その接続で許可されるトラフィックの量の上限を KB 単位で設定します。この値を超えると FXOS はセッションを切断します。

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

例：

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

ステップ 9 サーバのキー再生成の時間制限について、SSHセッションがアイドル状態を続けられる時間の上限を分単位で設定します。この値を超えると、FXOS はセッションを切断します。

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

例：

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

ステップ 10 トランザクションをシステム設定にコミットします。

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

ステップ 11 厳密なホスト キー チェックを設定して、SSH ホスト キーのチェックを制御します。

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

例：

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

- [enable]：FXOS が認識するホスト ファイルにそのホスト キーがまだ存在しない場合、接続は拒否されます。 FXOS CLI でシステム スコープまたはサービス スコープの **enter ssh-host** コマンドを使用して、手動でホストを追加する必要があります。
- [プロンプト（prompt）]：シャーシにまだ保存されていないホストキーを許可または拒否するように求められます。
- **disable**：（デフォルト）シャーシは過去に保存されたことがないホストキーを自動的に許可します。

ステップ 12 クライアントの暗号化アルゴリズムを設定します。

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

例：

```
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
```

（注）
- コモン クライテリアでは3des-cbcがサポートされていません。FXOSシャーシでコモンクライテリアモードが有効な場合、暗号化アルゴリズムとして3des-cbcを使用することはできません。
- 次の暗号化アルゴリズムは、デフォルトでは有効になっていません。

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

ステップ 13 クライアントの Diffie-hellman（DH）キー交換アルゴリズムを設定します。

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

例：

```
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
```

DH キー交換では、いずれの当事者も単独では決定できない共有秘密を使用します。キー交換を署名およびホストキーと組み合わせることで、ホスト認証が実現します。このキー交換方式により、明示的なサーバ認証が可能となります。DH キー交換の使用方法の詳細については、RFC 4253 を参照してください。

ステップ 14 クライアントの MAC アルゴリズムを設定します。

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

例：

```
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
```

ステップ 15 クライアントの ホスト キーについて、RSA キー ペアのモジュラス サイズを入力します。

モジュラス値（ビット単位）は、1024 ～ 2048 の範囲内の 8 の倍数です。指定するキー係数のサイズが大きいほど、RSA キー ペアの生成にかかる時間は長くなります。値は 2048 にすることをお勧めします。

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

例：

```
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
```

ステップ 16 クライアントのキー再生成のボリューム制限について、その接続で許可されるトラフィックの量の上限を KB 単位で設定します。この値を超えると FXOS はセッションを切断します。

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

例：

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

ステップ 17 クライアントのキー再生成の時間制限について、SSHセッションがアイドル状態を続けられる時間の上限を分単位で設定します。この値を超えると、FXOS はセッションを切断します。

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

例：

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

ステップ 18 トランザクションをシステム設定にコミットします。

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

例

次の例では、シャーシへの SSH アクセスを有効化し、トランザクションをコミットします。

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

TLS の設定

Transport Layer Security（TLS）プロトコルは、互いに通信する 2 つのアプリケーションの間でプライバシーとデータの整合性を確保します。FXOS シャーシと外部デバイスとの通信で許容する最小 TLS バージョンは、FXOS CLI を使用して設定できます。新しいバージョンの TLS では通信のセキュリティを強化できる一方、古いバージョンの TLS では古いアプリケーションとの後方互換性を維持できます。

たとえば、FXOS シャーシで設定されている最小 TLS バージョンが v1.1 の場合、クライアントブラウザが v1.0 だけを実行するように設定されていると、クライアントは HTTPS を使用して FXOS Chassis Manager との接続を開くことができません。したがって、ピア アプリケーションと LDAP サーバを適切に設定する必要があります。

次の手順で、FXOS シャーシと外部デバイス間の通信で許容する最小 TSL バージョンを設定、表示する方法を説明します。

（注）
- FXOS 2.3(1) リリースの時点では、FXOS シャーシのデフォルト最小 TLS バージョンは v1.1 です。

手順

ステップ 1 システム モードに入ります。

Firepower-chassis# **scope system**

ステップ 2 システムで使用できる TLS バージョンのオプションを表示します。

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

例：

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

ステップ 3 最小 TLS バージョンを設定します。

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

例：

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

ステップ 4 設定をコミットします。

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

ステップ 5 システムで設定されている最小 TLS バージョンを表示します。

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

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

例：

```
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
```

Telnet の設定

次の手順では、シャーシへの Telnet アクセスを有効化または無効化にする方法について説明します。デフォルトでは、Telnet は無効化になっています。

（注） 現在、Telnet は CLI を使用してのみ設定できます。

手順

ステップ 1 システム モードに入ります。

Firepower-chassis # **scope system**

ステップ 2 システム サービス モードを開始します。

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

ステップ 3 シャーシへの Telnet アクセスを設定するには、次のいずれかを実行します。

- シャーシへの Telnet アクセスを許可するには、次のコマンドを入力します。

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

- シャーシへの Telnet アクセスを禁止するには、次のコマンドを入力します。

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

ステップ 4 トランザクションをシステム設定にコミットします。

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

例

次に、Telnet を有効にし、トランザクションを確定する例を示します。

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

SNMP の設定

このセクションでは、シャーシに Simple Network Management Protocol（SNMP）を設定する方法を説明します。詳細については、次のトピックを参照してください。

SNMP の概要

簡易ネットワーク管理プロトコル（SNMP）は、SNMP マネージャとエージェント間の通信用メッセージフォーマットを提供する、アプリケーションレイヤプロトコルです。SNMP では、ネットワーク内のデバイスのモニタリングと管理に使用する標準フレームワークと共通言語が提供されます。

SNMP フレームワークは 3 つの部分で構成されます。

- SNMP マネージャ：SNMP を使用してネットワークデバイスのアクティビティを制御し、モニタリングするシステム
- SNMP エージェント：シャーシのデータを維持し、必要に応じてそのデータを SNMP マネージャに報告するシャーシ内のソフトウェアコンポーネント。シャーシには、エージェントと一連の MIB が含まれています。SNMP エージェントを有効にし、マネージャとエージェント間のリレーションシップを作成するには、Firepower Chassis Manager または FXOS CLI で SNMP を有効にし、設定します。
- 管理情報ベース（MIB）：SNMP エージェント上の管理対象オブジェクトのコレクション。

シャーシは、SNMPv1、SNMPv2c、および SNMPv3 をサポートします。SNMPv1 および SNMPv2c はどちらも、コミュニティベース形式のセキュリティを使用します。SNMP は次のように定義されています。

- 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）

（注） SNMP バージョン 1 および 2c には、重大な既知のセキュリティ問題があるので注意してください。これらのバージョンでは、すべての情報が暗号化されずに送信されます。これらのバージョンで唯一の認証形式として機能するコミュニティストリングも含まれます。

SNMP 通知

SNMP の重要な機能の 1 つは、SNMP エージェントから通知を生成できることです。これらの通知では、要求を SNMP マネージャから送信する必要はありません。通知は、不正なユーザ認証、再起動、接続の切断、隣接ルータとの接続の切断、その他の重要なイベントを表示します。

シャーシは、トラップまたはインフォームとして SNMP 通知を生成します。SNMP マネージャはトラップ受信時に確認応答を送信せず、シャーシはトラップが受信されたかどうかを確認できないため、トラップの信頼性はインフォームよりも低くなります。インフォーム要求を受信する SNMP マネージャは、SNMP 応答プロトコル データ ユニット（PDU）でメッセージの受信を確認応答します。シャーシが PDU を受信しない場合、インフォーム要求を再送できます。

ただし、インフォームは SNMPv2c でのみ使用可能ですが、安全ではないと考えられているため、推奨されません。

（注） SNMP を使用するインターフェイスの ifindex の順序は、FXOS の再起動後も変更されません。ただし、FXOS ディスク使用率 OID のインデックス番号は、FXOS を再起動すると変更されます。

SNMP セキュリティ レベルおよび権限

SNMPv1、SNMPv2c、および SNMPv3 はそれぞれ別のセキュリティ モデルを表します。セキュリティ モデルと選択したセキュリティ レベルの組み合わせにより、SNMP メッセージの処理中に適用されるセキュリティ メカニズムが決まります。

セキュリティ レベルは、SNMP トラップに関連付けられているメッセージを表示するために必要な特権を決定します。権限レベルは、メッセージが開示されないよう保護または認証の必要があるかどうかを決定します。サポートされるセキュリティ レベルは、セキュリティ モデルが設定されているかによって異なります。SNMP セキュリティ レベルは、次の権限の 1 つ以上をサポートします。

- noAuthNoPriv：認証なし、暗号化なし

- authNoPriv：認証あり、暗号化なし
- authPriv：認証あり、暗号化あり

SNMPv3 では、セキュリティ モデルとセキュリティ レベルの両方が提供されています。セキュリティ モデルは、ユーザおよびユーザが属するロールを設定する認証方式です。セキュリティ レベルとは、セキュリティ モデル内で許可されるセキュリティのレベルです。セキュリティ モデルとセキュリティ レベルの組み合わせにより、SNMP パケット処理中に採用されるセキュリティ メカニズムが決まります。

SNMP セキュリティ モデルとレベルのサポートされている組み合わせ

次の表に、セキュリティ モデルとレベルの組み合わせの意味を示します。

表 *1: SNMP* セキュリティ モデルおよびセキュリティ レベル

モデル	水準器	認証	暗号化	結果
v1	noAuthNoPriv	コミュニティ ストリング（Community string）	なし	コミュニティ ストリングの照合を使用して認証します。
v2c	noAuthNoPriv	コミュニティ ストリング（Community string）	なし	コミュニティ ストリングの照合を使用して認証します。
v3	noAuthNoPriv	[ユーザ名（Username）]	なし	ユーザ名の照合を使用して認証します。 （注） 設定することはできますが、FXOS では SNMP バージョン 3 で `noAuthNoPriv` を使用することはできません。
v3	authNoPriv	HMAC-SHA	なし	HMAC Secure Hash Algorithm（SHA）に基づいて認証します。
v3	authPriv	HMAC-SHA	DES	HMAC-SHA アルゴリズムに基づいて認証します。データ暗号規格（DES）の 56 ビット暗号化、および暗号ブロック連鎖（CBC）DES（DES-56）標準に基づいた認証を提供します。

SNMPv3 セキュリティ機能

SNMPv3 は、ネットワーク経由のフレームの認証と暗号化を組み合わせることによって、デバイスへのセキュア アクセスを実現します。SNMPv3 は、設定済みユーザによる管理動作のみを

許可し、SNMP メッセージを暗号化します。SNMPv3 ユーザベース セキュリティ モデル（USM）は SNMP メッセージレベル セキュリティを参照し、次のサービスを提供します。

- メッセージの完全性：メッセージが不正な方法で変更または破壊されていないことを保証します。また、データ シーケンスが、通常発生するものよりも高い頻度で変更されていないことを保証します。
- メッセージ発信元の認証：受信データを発信したユーザのアイデンティティが確認されたことを保証します。
- メッセージの機密性および暗号化：不正なユーザ、エンティティ、プロセスに対して情報を利用不可にしたり開示しないようにします。

SNMP サポート

シャーシは、SNMP に次のサポートを提供します。

MIB のサポート

シャーシは、MIB への読み取り専用アクセスをサポートします。

利用可能な特定の MIB の詳細とその入手場所については、『Cisco FXOS MIB Reference Guide』を参照してください。

SNMPv3 ユーザの認証プロトコル

シャーシは、SNMPv3 ユーザーの HMAC-SHA-96（SHA）認証プロトコルをサポートします。

SNMPv3 ユーザの AES プライバシー プロトコル

シャーシは、SNMPv3 メッセージ暗号化用プライバシープロトコルの 1 つとして、Advanced Encryption Standard（AES）を使用し、RFC 3826 に準拠します。

プライバシーパスワード（priv オプション）では、SNMP セキュリティ暗号化方式として DES または 128 ビット AES を選択できます。AES-128 の設定を有効にして、SNMPv3 ユーザー用のプライバシーパスワードを含めると、シャーシはそのプライバシーパスワードを使用して 128 ビット AES キーを生成します。AES priv パスワードは、8 文字以上にします。パスフレーズをクリア テキストで指定する場合、最大 64 文字を指定できます。

SNMP の有効化および SNMP プロパティの設定

手順

ステップ 1 スコープモードに入ります。

Firepower-chassis# **scope ssa**

ステップ 2 show app-instance コマンドを入力し、スロット ID、アプリケーション名、およびアプリケーション インスタンスの識別子を確認します。

Firepower-chassis# **show app-instance**

ステップ 3 モニタリング モードを開始します。

Firepower-chassis# **scope monitoring**

ステップ 4 （オプション）ASA デバイスと FTD デバイスの SNMP 管理アプリインスタンスモードを開始します。

Firepower-chassis /monitoring # **set snmp adminappinstance slot 1 appname ftd id ftd1 enable yes**

スロット番号、アプリ名、ID を指定し、有効化を [はい（Yes）] または [いいえ（No）] に設定して、ターゲット ブレード アプリ インスタンスを指定する必要があります。

重要 SNMP 統合を設定したら、5 分間待ってから、SNMP ポーリングに進みます。

ステップ 5 （任意）SNMP コミュニティモードを開始します。

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

set snmp community コマンドを入力すると、SNMP コミュニティ名の入力を求められます。

SNMP コミュニティ名を指定すると、SNMP リモートマネージャからのポーリング要求に対して SNMP バージョン 1 および 2c も自動的に有効になります。

（注） SNMP バージョン 1 および 2c には、重大な既知のセキュリティ問題があるので注意してください。これらのバージョンでは、すべての情報が暗号化されずに送信されます。これらのバージョンで唯一の認証形式として機能するコミュニティストリングも含まれます。

ステップ 6 SNMP コミュニティ名を指定します。このコミュニティ名は、SNMP パスワードとして使用されます。コミュニティ名は、最大 32 文字の英数字で指定できます。

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

コミュニティ名は 1 つだけです。ただし、**set snmp community** を使用して既存の名前を上書きすることができます。既存のコミュニティ名を削除する（SNMP リモート マネージャからのポーリング要求に対して SNMP バージョン 1 および 2c も無効にする）には、**set snmp community** を入力します。ただし、コミュニティストリングを入力しないでください。つまり、もう一度 **Enter** キーを押します。バッファをコミットすると、**show snmp** の出力に `Is Community Set: No` という行が含まれます。

ステップ 7 SNMP 担当者のシステムの連絡先を指定します。システムの連絡先名（電子メール アドレスや、名前と電話番号など）は、最大 255 文字の英数字で指定できます。

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

ステップ 8 SNMP エージェント（サーバ）が実行されるホストの場所を指定します。システム ロケーション名は、最大 512 文字の英数字で指定できます。

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

ステップ 9 トランザクションをシステム設定にコミットします。

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

例

次に、SNMP を有効にし、SnmpCommSystem2 という名前の SNMP コミュニティを設定し、contactperson という名前のシステム連絡先を設定し、systemlocation という名前の連絡先ロケーションを設定し、トランザクションをコミットする例を示します。

```
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 # 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 #
```

次のタスク

SNMP トラップおよびユーザを作成します。

SNMP トラップの作成

次の手順では、SNMP トラップを作成する方法について説明します。

（注） 最大 8 つの SNMP トラップを定義できます。

手順

ステップ 1 モニタリング モードを開始します。

Firepower-chassis# **scope monitoring**

ステップ 2 SNMP をイネーブルにします。

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

ステップ 3 指定したホスト名、IPv4 アドレス、または IPv6 アドレスで SNMP トラップを作成します。

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

ステップ 4 SNMP トラップで使用する SNMP コミュニティストリングまたはバージョン 3 のユーザ名を指定します。

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

トラップの宛先へのアクセスを許可するために必要な SNMPv1/v2c コミュニティストリングまたは SNMPv3 ユーザ名を指定します。このコマンドを入力すると、コミュニティ名が照会されます。名前は最大 32 文字で、スペースは使用できません。名前は入力しても表示されません。

ステップ 5 SNMP トラップに使用するポートを指定します。

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

ステップ 6 トラップに使用する SNMP のバージョンとモデルを指定します。

Firepower-chassis /monitoring/snmp-trap # **set version** {**v1** | **v2c** | **v3**}

（注） SNMP バージョン 1 および 2c には、重大な既知のセキュリティ問題があるので注意してください。これらのバージョンでは、すべての情報が暗号化されずに送信されます。これらのバージョンで唯一の認証形式として機能するコミュニティストリングも含まれます。

ステップ 7 （任意） 送信するトラップのタイプを指定します。

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

ここに表示される値は次のとおりです。

- バージョンに [v2c] または [v3] を選択する場合は **traps**。
- バージョンに v2c を選択する場合は **informs**。

（注） バージョンに v2c を選択した場合のみ、インフォーム通知を送信できます。

ステップ 8 （任意） バージョンで v3 を選択した場合は、トラップに関連付ける権限を指定します。

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

ここに表示される値は次のとおりです。

- [auth]：認証あり、暗号化なし
- [noauth]：認証なし、暗号化なし これを指定することはできますが、FXOS は SNMPv3 でこのセキュリティレベルをサポートしていないことに注意してください。
- [priv]：認証あり、暗号化あり

ステップ 9 トランザクションをシステム設定にコミットします。

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

例

次の例は、SNMP を有効にし、IPv4 アドレスを使用して SNMP トラップを作成し、トラップがポート 2 で SnmpCommSystem2 コミュニティを使用するよう指定し、バージョンを v3 に設定し、通知タイプを traps に設定し、v3 権限を priv に設定し、トランザクションをコミットします。

```
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 #
```

次の例は、SNMP を使用可能にし、IPv6 アドレスを使用して SNMP トラップを作成し、トラップがポート 2 で SnmpCommSystem3 コミュニティを使用するよう指定し、バージョンを v3 に設定し、通知タイプを traps に設定し、v3 権限を priv に設定し、トランザクションを確定します。

```
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 #
```

SNMP トラップの削除

手順

ステップ 1 モニタリング モードを開始します。

Firepower-chassis# **scope monitoring**

ステップ 2 指定したホスト名または IP アドレスの SNMP トラップを削除します。

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

ステップ 3 トランザクションをシステム設定にコミットします。

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

例

次に、IP アドレス 192.168.100.112 で SNMP トラップを削除し、トランザクションをコミットする例を示します。

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

SNMPv3 ユーザの作成

手順

ステップ 1 モニタリング モードを開始します。

Firepower-chassis# **scope monitoring**

ステップ 2 SNMP をイネーブルにします。

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

ステップ 3 SNMPv3 ユーザを作成します。

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

create snmp-user コマンドを入力すると、パスワードの入力を促すプロンプトが表示されます。

FXOS では、次の要件を満たさないパスワードは拒否されます。

- 8 ～ 80 文字を含む。
- 含められるのは、文字、数字、および次の文字のみです。

  ~`!@#%^&*()_-+{}[]|\:;"'<,>./
- 次の記号を含まない。$（ドル記号）、?（疑問符）、「=」（等号）。
- 5 つ以上の異なる文字を含める必要があります。
- 連続するインクリメントまたはデクリメントの数字または文字をたくさん含めないでください。たとえば、「12345」 は 4 つ、「ZYXW」は 3 つ文字列が続いています。このような文字の合計数が特定の制限を超えると（通常は約 4 ～ 6 回発生）、簡素化チェックに失敗します。

（注） 連続するインクリメントまたはデクリメント文字列の間に連続しないインクリメントまたはデクリメント文字列が含まれても、文字数はリセットされません。たとえば、abcd&!21 はパスワード チェックに失敗しますが、abcd&!25 は失敗しません。

ステップ 4 SHA 認証の使用を指定します。

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

ステップ 5 AES-128 暗号化の使用を有効化またはディセーブルにします。

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

デフォルトでは、AES-128 暗号化はディセーブルになっています。

SNMPv3 は DES をサポートしていません。AES-128 を無効のままにすると、プライバシーの暗号化は行われず、設定されたプライバシーパスワードは無効になります。

（注） SNMPv3 が Authpriv（DES）で有効になっている場合、特定の NMS モニタリング アプリケーションから SNMPv3 FXOS デバイスをポーリングできません。以前に DES の使用をサポートしていたバージョンからデバイスをアップグレードする場合は、AES を使用してユーザーを再作成し、SNMPv3 FXOS デバイスをポーリングする必要があります。

ステップ 6 ユーザーパスワードを指定します。

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

set password コマンドを入力すると、パスワードの入力と確認を促すプロンプトが表示されます。

ステップ 7 トランザクションをシステム設定にコミットします。

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

例

次の例では、SNMP を有効化し、snmp-user14 という名前の SNMPv3 ユーザを作成し、AES-128 暗号化を有効化し、パスワードおよびプライバシー パスワードを設定し、トランザクションをコミットします。

```
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 #
```

SNMPv3 ユーザの削除

手順

ステップ 1 モニタリング モードを開始します。

Firepower-chassis# **scope monitoring**

ステップ 2 指定した SNMPv3 ユーザを削除します。

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

ステップ 3 トランザクションをシステム設定にコミットします。

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

例

次に、snmp user14 という名前の SNMPv3 ユーザを削除し、トランザクションをコミットする例を示します。

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

現在の SNMP 設定の表示

現在の SNMP 設定、ユーザ、およびトラップを表示するには、次の CLI コマンドを使用します。

（注） SNMP を使用する FXOS のインターフェイスの ifIndex の順序は、FXOS の再起動後も変更されません。

手順

ステップ 1 モニタリング モードを開始します。

firepower# **scope monitoring**

ステップ 2 現在の SNMP 設定を表示します。

firepower/monitoring # **show snmp**

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

ステップ 3 現在定義されている SNMPv3 ユーザを一覧表示します。

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

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

ステップ 4 現在定義されている SNMP トラップを一覧表示します。

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
```

例

次に、特定の SNMPv3 ユーザに関する詳細情報を表示する例を示します。

```
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 #
```

HTTPS の設定

ここでは、Firepower 4100/9300 シャーシ で HTTPS を設定する方法を説明します。

（注） Firepower Chassis Manager または FXOS CLI を使用して HTTPS ポートを変更できます。他の HTTPS の設定はすべて、FXOS CLI を使用してのみ設定できます。

証明書、キー リング、トラスト ポイント

HTTPS は、公開キー インフラストラクチャ（PKI）を使用してクライアントのブラウザと Firepower 4100/9300 シャーシ などの 2 つのデバイス間でセキュアな通信を確立します。

暗号キーとキー リング

各 PKI デバイスは、内部キー リングに非対称の Rivest-Shamir-Adleman（RSA）暗号キーのペア（1 つはプライベート、もう 1 つはパブリック）を保持します。いずれかのキーで暗号化されたメッセージは、もう一方のキーで復号化できます。暗号化されたメッセージを送信する場合、送信者は受信者の公開キーで暗号化し、受信者は独自の秘密キーを使用してメッセージを復号化します。送信者は、独自の秘密キーで既知のメッセージを暗号化（「署名」とも呼ばれます）して公開キーの所有者を証明することもできます。受信者が該当する公開キーを使用してメッセージを正常に復号化できる場合は、送信者が対応する秘密キーを所有していることが証明されます。暗号キーの長さはさまざまであり、通常の長さは 512 ビット ～ 2048 ビットです。通常、長いキーは短いキーよりもより安全です。FXOS では最初に 2048 ビットのキー ペアを含むデフォルトのキー リングが提供されます。そして、追加のキー リングを作成できます。

クラスタ名が変更されたり、証明書が期限切れになったりした場合は、デフォルトのキー リング証明書を手動で再生成する必要があります。

証明書

セキュアな通信を準備するには、まず2つのデバイスがそれぞれのデジタル証明書を交換します。証明書は、デバイスの ID に関する署名済み情報とともにデバイスの公開キーを含むファイルです。暗号化された通信をサポートするために、デバイスは独自のキーペアと独自の自己署名証明書を生成できます。リモート ユーザが自己署名証明書を提示するデバイスに接続する場合、ユーザはデバイスの ID を簡単に検証することができず、ユーザのブラウザは最初に認証に関する警告を表示します。デフォルトでは、FXOSにはデフォルトのキー リングからの公開キーを含む組み込みの自己署名証明書が含まれます。

トラスト ポイント

FXOS に強力な認証を提供するために、デバイスの ID を証明する信頼できるソース（つまり、トラスト ポイント）からサードパーティ証明書を取得し、インストールできます。サードパーティ証明書は、発行元トラスト ポイント（ルート認証局（CA）、中間 CA、またはルート CA につながるトラスト チェーンの一部となるトラスト アンカーのいずれか）によって署名されます。新しい証明書を取得するには、FXOS で証明書要求を生成し、トラスト ポイントに要求を送信する必要があります。

重要 証明書は、Base64 エンコード X.509（CER）フォーマットである必要があります。

キー リングの作成

FXOS は、デフォルト キー リングを含め、最大 8 個のキー リングをサポートします。

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis # **scope security**

ステップ 2 キー リングを作成し、名前を付けます。

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

ステップ 3 SSL キーのビット長を設定します。

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

ステップ 4 トランザクションをコミットします。

Firepower-chassis # **commit-buffer**

例

次の例は、1024 ビットのキー サイズのキー リングを作成します。

```
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 #
```

次のタスク

このキー リングの証明書要求を作成します。

デフォルト キー リングの再生成

クラスタ名が変更されたり、証明書が期限切れになったりした場合は、デフォルトのキーリング証明書を手動で再生成する必要があります。

（注） デフォルトのキーリングは、FXOS 上の FCM によってのみ使用されます。

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis # **scope security**

ステップ 2 デフォルト キー リングでキー リング セキュリティ モードに入ります。

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

ステップ 3 デフォルト キー リングを再生成します。

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

ステップ 4 トランザクションをコミットします。

Firepower-chassis # **commit-buffer**

例

次に、デフォルト キー リングを再生成する例を示します。

```
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 #
```

キー リングの証明書要求の作成

基本オプション付きのキー リングの証明書要求の作成

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis # **scope security**

ステップ 2 キー リングのコンフィギュレーション モードに入ります。

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

ステップ 3 指定された IPv4 または IPv6 アドレス、またはファブリック インターコネクトの名前を使用して証明書要求を作成します。証明書要求のパスワードを入力するように求められます。

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

ステップ 4 トランザクションをコミットします。

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

ステップ 5 コピーしてトラスト アンカーまたは認証局に送信可能な証明書要求を表示します。

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

例

次の例では、基本オプション付きのキー リングについてIPv4アドレスで証明書要求を作成して表示します。

```
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 #
```

次のタスク

- 証明書要求のテキストを BEGIN および END 行を含めてコピーし、ファイルに保存します。キー リングの証明書を取得するため、証明書要求を含むファイルをトラストアンカーまたは認証局に送信します。
- トラスト ポイントを作成し、トラスト アンカーから受け取ったトラストの証明書の証明書チェーンを設定します。

詳細オプション付きのキー リングの証明書要求の作成

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis # **scope security**

ステップ 2 キー リングのコンフィギュレーション モードに入ります。

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

ステップ 3 証明書要求を作成します。

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

ステップ 4 会社が存在している国の国コードを指定します。

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

ステップ 5 要求に関連付けられたドメイン ネーム サーバ（DNS）アドレスを指定します。

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

ステップ 6 証明書要求に関連付けられた電子メール アドレスを指定します。

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

ステップ 7 Firepower 4100/9300 シャーシの IP アドレスを指定します。

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

ステップ 8 証明書を要求している会社の本社が存在する市または町を指定します。

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

ステップ 9 証明書を要求している組織を指定します。

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

ステップ 10 組織ユニットを指定します。

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

ステップ 11 証明書要求に関するオプションのパスワードを指定します。

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

ステップ 12 証明書を要求している会社の本社が存在する州または行政区分を指定します。

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

ステップ 13 Firepower 4100/9300 シャーシ の完全修飾ドメイン名を指定します。

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

ステップ 14 トランザクションをコミットします。

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

ステップ 15 コピーしてトラスト アンカーまたは認証局に送信可能な証明書要求を表示します。

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

例

（注） 2.7 より前のリリースでは、「set dns」または「set subject-name」で FQDN を使用せずにバッファをコミットすることはお勧めできません。FQDN ではない DNS またはサブジェクト名を使用して認証要件を作成しようとすると、エラーがスローされます。

次の例では、詳細オプション付きのキー リングについて IPv4 アドレスで証明書要求を作成して表示します。

```
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 #
```

次のタスク

- 証明書要求のテキストを BEGIN および END 行を含めてコピーし、ファイルに保存します。キーリングの証明書を取得するため、証明書要求を含むファイルをトラストアンカーまたは認証局に送信します。
- トラスト ポイントを作成し、トラスト アンカーから受け取ったトラストの証明書の証明書チェーンを設定します。

トラスト ポイントの作成

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis # **scope security**

ステップ 2 トラストポイントを作成します。

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

ステップ 3 このトラスト ポイントの証明書情報を指定します。

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

コマンドで証明書情報を指定しない場合、ルート認証局（CA）への認証パスを定義するトラスト ポイントのリストまたは証明書を入力するように求められます。入力内容の次の行に、**ENDOFBUF** と入力して終了します。

重要 証明書は、Base64 エンコード X.509（CER）フォーマットである必要があります。

ステップ 4 トランザクションをコミットします。

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

例

次の例は、トラスト ポイントを作成し、トラスト ポイントに証明書を提供します。

```
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
> BxMMU2FulEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGxlIEluYy4xEzARBgNVBAsT
> 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 #
```

次のタスク

トラスト アンカーまたは認証局からキー リング証明書を取得し、キー リングにインポートします。

キー リングへの証明書のインポート

始める前に

- キー リング証明書の証明書チェーンを含むトラスト ポイントを設定します。
- トラスト アンカーまたは認証局からキー リング証明書を取得します。

（注） HTTPS ですでに設定されているキーリングの証明書を変更する場合は、新しい証明書を有効にするために HTTPS を再起動する必要があります。詳細については、HTTPS の再起動 （39 ページ） を参照してください。

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis # **scope security**

ステップ 2 証明書を受け取るキー リングでコンフィギュレーション モードに入ります。

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

ステップ 3 キー リング証明書の取得元のトラスト アンカーまたは認証局に対しトラスト ポイントを指定します。

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

ステップ 4 キー リング証明書を入力してアップロードするためのダイアログを起動します。

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

プロンプトで、トラストアンカーまたは認証局から受け取った証明書のテキストを貼り付けます。証明書の後の行に **ENDOFBUF** と入力して、証明書の入力を完了します。

重要 証明書は、Base64 エンコード X.509（CER）フォーマットである必要があります。

ステップ 5 トランザクションをコミットします。

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

例

次に、トラストポイントを指定し、証明書をキー リングにインポートする例を示します。

```
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 #
```

次のタスク

キー リングを使用して HTTPS サービスを設定します。

HTTPS の設定

⚠

注意 HTTPS で使用するポートとキー リングの変更を含め、HTTPS の設定を完了した後、トランザクションを保存またはコミットするとすぐに、現在のすべての HTTP および HTTPS セッションは警告なく閉じられます。

手順

ステップ 1 システム モードに入ります。

Firepower-chassis# **scope system**

ステップ 2 システム サービス モードを開始します。

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

ステップ 3 HTTPS サービスを有効にします。

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

ステップ 4 （任意） HTTPS 接続で使用されるポートを指定します。

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

ステップ 5 （任意） HTTPS に対して作成したキー リングの名前を指定します。

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

ステップ 6 （任意） ドメインで使用される暗号スイート セキュリティのレベルを指定します。

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

cipher-suite-mode には、以下のいずれかのキーワードを指定できます。

- **high-strength**
- **medium-strength**
- **low-strength**
- **custom**：ユーザ定義の暗号スイート仕様の文字列を指定できます。

ステップ 7 （任意） **cipher-suite-mode** が **custom** に設定されている場合は、ドメインに対してカスタム レベルの暗号スイート セキュリティを指定します。

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

cipher-suite-spec-string は最大 256 文字で構成できます。これは OpenSSL 暗号スイート仕様に準拠する必要があります。次を除き、スペースや特殊文字は使用できません。！（感嘆符）、+（プラス記号）、-（ハイフン）、および :（コロン）。詳細については、http://httpd.apache.org/docs/2.0/mod/mod_ssl.html#sslciphersuite を参照してください。

たとえば、FXOS がデフォルトとして使用する中強度仕様の文字列は次のようになります。
`ALL:!ADH:!EXPORT56:!LOW:RC4+RSA:+HIGH:+MEDIUM:+EXP:+eNULL`

（注） **cipher-suite-mode** は **custom** 以外に設定されている場合、このオプションは無視されます。

ステップ 8 （任意） 証明書失効リスト検査を、有効または無効にします。

set revoke-policy *{ relaxed | strict }*

ステップ 9 トランザクションをシステム設定にコミットします。

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

例

次の例では、HTTPS をイネーブルにし、ポート番号を 443 に設定し、キーリング名を kring7984 に設定し、暗号スイートのセキュリティ レベルを [high] に設定し、トランザクションをコミットします。

```
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 #
```

HTTPS ポートの変更

HTTPS サービスは、デフォルトでポート 443 で有効化になります。HTTPS をディセーブルにすることはできませんが、HTTPS 接続に使用するポートは変更できます。

手順

ステップ 1 システム モードに入ります。

Firepower-chassis # **scope system**

ステップ 2 システム サービス モードを開始します。

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

ステップ 3 HTTPS 接続に使用するポートを指定します。

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

port-number には 1 ～ 65535 の整数を指定します。HTTPS は、デフォルトではポート 443 で有効になっています。

ステップ 4 トランザクションをシステム設定にコミットします。

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

HTTPS ポートを変更した後に、現在のすべての HTTPS セッションが閉じられます。ユーザは、次のように新しいポートを使用して再度 Firepower Chassis Manager にログインする必要があります。

https://*<chassis_mgmt_ip_address>*:*<chassis_mgmt_port>*

<*chassis_mgmt_ip_address*> は、初期設定時に入力したシャーシの IP アドレスまたはホスト名で、<*chassis_mgmt_port*> は設定が完了した HTTPS ポートです。

例

次に、HTTPS ポート番号を 443 に設定し、トランザクションを確定する例を示します。

```
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 #
```

HTTPS の再起動

HTTPS ですでに設定されているキーリングの証明書を変更する場合は、新しい証明書を有効にするために HTTPS を再起動する必要があります。更新されたキーリングで HTTPS を再設定するには、次の手順を使用します。

手順

ステップ 1 システム モードに入ります。

Firepower-chassis# **scope system**

ステップ 2 システム サービス モードを開始します。

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

ステップ 3 HTTPS キーリングをデフォルト値に戻します。

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

ステップ 4 トランザクションをシステム設定にコミットします。

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

ステップ 5 5 秒間待機します。

ステップ 6 作成したキーリングで HTTPS を設定します。

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

ステップ 7 トランザクションをシステム設定にコミットします。

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

キーリングの削除

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis # **scope security**

ステップ 2 名前付きのキー リングを削除します。

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

ステップ 3 トランザクションをコミットします。

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

例

次の例では、キー リングを削除します。

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

トラスト ポイントの削除

始める前に

トラスト ポイントがキー リングによって使用されていないことを確認してください。

手順

ステップ 1 セキュリティ モードに入ります。

Firepower-chassis# **scope security**

ステップ 2 指定したトラスト ポイントを削除します。

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

ステップ 3 トランザクションをコミットします。

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

例

次に、トラスト ポイントを削除する例を示します。

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

HTTPS の無効化

手順

ステップ 1 システム モードに入ります。

Firepower-chassis# **scope system**

ステップ 2 システム サービス モードを開始します。

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

ステップ 3 HTTPS サービスを無効にします。

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

ステップ 4 トランザクションをシステム設定にコミットします。

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

例

次に、HTTPS を無効にし、トランザクションをコミットする例を示します。

```
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 #
```

AAA の設定

ここでは、認証、許可、およびアカウンティングについて説明します。詳細については、次のトピックを参照してください。

AAA について

認証、許可、およびアカウンティング（AAA）は、ネットワークリソースへのアクセス制御、ポリシーの強化、使用状況の評価、およびサービスの課金に必要な情報提供を行う一連のサービスです。認証は、ユーザを識別します。認可は、認証されたユーザがアクセスする可能性があるリソースとサービスを決定するポリシーを実装します。アカウンティングは、課金と分析に使用される時間とデータのリソースを追跡します。これらの処理は、効果的なネットワーク管理およびセキュリティにとって重要です。

認証

認証はユーザを識別する方法です。通常、ユーザが有効なユーザ名と有効なパスワードを入力すると、アクセスが許可されます。AAA サーバは、ユーザが入力したログイン情報とデータベースに保存されているユーザのログイン情報を比較します。ログイン情報が一致する場合、ユーザはネットワークへのアクセスが許可されます。クレデンシャルが一致しない場合は、認証は失敗し、ネットワーク アクセスは拒否されます。

シャーシへの管理接続を認証するように Firepower 4100/9300 シャーシ を設定できます。これには、次のセッションが含まれます。

- HTTPS
- SSH
- シリアル コンソール

認可

許可はポリシーを適用するプロセスです。どのようなアクティビティ、リソース、サービスに対するアクセス許可をユーザが持っているのかを判断します。ユーザは認証後にさまざまなタイプのアクセスやアクティビティを許可される可能性があります。

アカウンティング

アカウンティングは、アクセス時にユーザが消費したリソースを測定します。これには、システム時間またはセッション中にユーザが送受信したデータ量などが含まれます。アカウンティングは、許可制御、課金、トレンド分析、リソース使用率、キャパシティプランニングのアクティビティに使用されるセッションの統計情報と使用状況情報のログを通じて行われます。

認証、認可、アカウンティング間の相互作用

認証は、単独で使用することも、認可およびアカウンティングとともに使用することもできます。認可では必ず、ユーザの認証が最初に済んでいる必要があります。アカウンティングだけで使用することも、認証および認可とともに使用することもできます。

サポートされている認証タイプ

FXOS は次の認証タイプをサポートします。

- [Remote]：次のネットワーク AAA サービスがサポートされています。

- LDAP
- RADIUS
- TACACS+

- [ローカル（Local）]：シャーシは、ユーザープロファイルを取り込むことができるローカルデータベースを維持します。 AAA サーバの代わりに、このローカルデータベースを使用して、ユーザ認証、認可、アカウンティングを提供することもできます。

ユーザ ロール

FXOS は、ユーザロール割り当ての形式でローカルおよびリモート認証をサポートします。割り当てることができるロールは次のとおりです。

- [Admin]：システム全体に対する完全な読み取りと書き込みのアクセス権。デフォルトの admin アカウントは、デフォルトでこのロールが割り当てられ、変更はできません。
- [AAA Administrator]：ユーザ、ロール、および AAA 設定に対する読み取りと書き込みのアクセス権。システムの残りの部分に対する読み取りアクセス権。
- [Operations]：NTP の設定、Smart Licensing のための Smart Call Home の設定、システム ログ（syslog サーバとエラーを含む）に対する読み取りと書き込みのアクセス権。システムの残りの部分に対する読み取りアクセス権。
- [Read-Only]：システム設定に対する読み取り専用アクセス権。システム状態を変更する権限はありません。

ローカル ユーザとロールの割り当ての詳細については、「User Management」を参照してください。

AAA の設定

Firepower 4100/9300 アプライアンスで認証、許可、アカウンティング（AAA）を設定するための基本的な手順の概要を紹介します。

1. ユーザ認証の目的タイプを設定します。
   - [Local]：ユーザ定義とローカル認証は User Management の一部です。
   - [Remote]：リモート AAA サーバ アクセスの設定は、[Platform Settings] の一部です。具体的には次のとおりです。
     - LDAP プロバイダーの設定 （44 ページ）
     - RADIUS プロバイダーの設定 （49 ページ）
     - TACACS+ プロバイダーの設定 （52 ページ）

（注） リモート AAA サーバーを使用する場合は、シャーシでリモート AAA サーバーアクセスを設定する前に、リモートサーバーで AAA サービスを有効にして設定する必要があります。

2. デフォルトの認証方式を指定します。これも User Management の一部です。

（注） デフォルトの認証とコンソール認証の両方が同じリモート認証プロトコル（RADIUS、TACACS+、または LDAP）を使用するように設定されている場合、そのサーバの設定の特定の側面を変更することは（たとえば、サーバの削除や、割り当ての順序の変更）、これらのユーザ設定を更新することなしではできません。

LDAP プロバイダーの設定

LDAP プロバイダーのプロパティの設定

このタスクで設定するプロパティが、このタイプのすべてのプロバイダー接続のデフォルト設定です。個々のプロバイダーにいずれかのプロパティの設定が含まれている場合、FXOS でその設定が使用され、デフォルト設定は無視されます。

Active Directory を LDAP サーバとして使用している場合は、Active Directory サーバで FXOS にバインドするユーザアカウントを作成します。このアカウントには、期限切れにならないパスワードを設定します。

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis# **scope security**

ステップ 2 セキュリティ LDAP モードを開始します。

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

ステップ 3 指定した属性を含むレコードにデータベース検索を限定します。

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

ステップ 4 指定した識別名を含むレコードにデータベース検索を限定します。

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

ステップ 5 指定したフィルタを含むレコードにデータベース検索を限定します。

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

ここで、*filter* は LDAP サーバで使用するフィルタ属性です（*cn = $userid*、*sAMAccountName = $userid* など）。フィルタには *$userid* が含まれている必要があります。

ステップ 6 システムがサーバをダウン状態として通知する前に、LDAP サーバからの応答を待つ時間を設定します。

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

ステップ 7 トランザクションをシステム設定にコミットします。

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

例

次の例では、LDAP 属性を CiscoAvPair に、ベース識別名を「DC=cisco-firepower-aaa3,DC=qalab,DC=com」に、フィルタを sAMAccountName=$userid に、タイムアウト間隔を 5 秒に設定し、トランザクションをコミットします。

```
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 #
```

（注） ユーザログインは、LDAP ユーザの DN が 255 文字を超えると失敗します。

次のタスク

LDAP プロバイダーを作成します。

LDAP プロバイダーの作成

次の手順に従い、LDAP プロバイダー（このアプライアンスに LDAP ベースの AAA サービスを提供する特定のリモートサーバー）を定義および設定します。

（注） FXOS では、最大 16 の LDAP プロバイダーをサポートします。

始める前に

Active Directory を LDAP サーバとして使用している場合は、Active Directory サーバで FXOS にバインドするユーザアカウントを作成します。このアカウントには、期限切れにならないパスワードを設定します。

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis# **scope security**

ステップ 2 セキュリティ LDAP モードを開始します。

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

ステップ 3 LDAP サーバ インスタンスを作成し、セキュリティ LDAP サーバ モードを開始します。

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

SSL が有効の場合、 *server-name* は、通常 IP アドレスまたは FQDN となり、LDAP サーバのセキュリティ証明書内の Common Name（CN）と正確に一致している必要があります。IP アドレスが指定されている場合を除き、DNS サーバを設定する必要があります。

ステップ 4 （任意） ユーザ ロールとロケールの値を保管する LDAP 属性を設定します。

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

このプロパティは、常に、名前と値のペアで指定されます。システムは、ユーザ レコードで、この属性名と一致する値を検索します。

デフォルトの属性が LDAP プロバイダー用に設定されていない場合は、この値が必要です。

ステップ 5 （任意）リモートユーザがログインし、システムがそのユーザ名に基づいてユーザの DN の取得を試みるときに、サーバが検索を開始する LDAP 階層内の特定の識別名を設定します。

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

ベース DN の長さは、最大 255 文字から CN=username の長さを引いた長さに設定することができます。username により、LDAP 認証を使用して Firepower Chassis Manager または FXOS CLI にアクセスしようとするリモートユーザが識別されます。

デフォルトのベース DN が LDAP プロバイダー用に設定されていない場合は、この値が必要です。

ステップ 6 （任意）ベース DN 下のすべてのオブジェクトに対する読み取り権限と検索権限を持つ、LDAP データベース アカウントの識別名（DN）を設定します。

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

サポートされるストリングの最大長は 255 文字（ASCII）です。

ステップ 7 （任意） LDAP 検索を、定義されたフィルタと一致するユーザ名に制限します。

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

ここで、*filter-value* はLDAPサーバで使用するフィルタ属性です（*cn = $userid*、*sAMAccountName = $userid* など）。フィルタには *$userid* が含まれている必要があります。

デフォルトのフィルタが LDAP プロバイダー用に設定されていない場合は、この値が必要です。

ステップ 8 バインド DN で指定した LDAP データベース アカウントのパスワードを指定します。

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

パスワードを設定するには、 **set password** コマンドを入力してから **Enter** を押し、プロンプトでキー値を入力します。

標準ASCII文字を入力できます。ただし、「§」（セクション記号）、「?」（疑問符）、「=」（等号）は除きます。

ステップ 9 （任意） FXOS でこのプロバイダーをユーザの認証に使用する順序を指定します。

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

ステップ 10 （任意） LDAP サーバとの通信に使用するポートを指定します。標準ポート番号は389です。

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

ステップ 11 LDAP サーバと通信するときの暗号化の使用を有効化またはディセーブルにします。

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

オプションは次のとおりです。

- **yes** ：暗号化が必要です。暗号化をネゴシエートできない場合は、接続に失敗します。
- **no** ：暗号化は無効です。認証情報はクリア テキストとして送信されます。

LDAP では STARTTLS が使用されます。これにより、ポート 389 を使用した暗号化通信が可能になります。

ステップ 12 LDAP データベースへの問い合わせがタイムアウトするまでの秒数を指定します。

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

1 ～ 60 秒の整数を入力するか、0（ゼロ）を入力して LDAP プロバイダーで指定したグローバル タイムアウト値を使用します。デフォルトは 30 秒です。

ステップ 13 LDAP プロバイダーやサーバの詳細を提供するベンダーを指定します。

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

オプションは次のとおりです。

- **ms-ad**：LDAP プロバイダーは Microsoft Active Directory です。
- **openldap**：LDAP プロバイダーは Microsoft Active Directory ではありません。

ステップ 14 （任意）証明書失効リスト検査を有効にします。

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

（注）　この設定は、SSL 接続が使用可能である場合にのみ有効です。

ステップ 15　トランザクションをシステム設定にコミットします。

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

例

次の例では、10.193.169.246 という名前の LDAP サーバインスタンスを作成し、binddn、パスワード、順序、ポート、SSL、ベンダー属性を設定し、トランザクションをコミットします。

```
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 #
```

次の例では、12:31:71:1231:45b1:0011:011:900 という名前の LDAP サーバインスタンスを作成し、バインドDN、パスワード、順序、ポート、SSL、ベンダー属性を設定し、トランザクションを確定します。

```
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 #
```

LDAP プロバイダーの削除

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis# **scope security**

ステップ 2 セキュリティ LDAP モードを開始します。

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

ステップ 3 指定したサーバを削除します。

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

ステップ 4 トランザクションをシステム設定にコミットします。

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

例

次に、ldap1という名前のLDAPサーバを削除し、トランザクションをコミットする例を示します。

```
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 #
```

RADIUS プロバイダーの設定

RADIUS プロバイダーのプロパティの設定

このタスクで設定するプロパティが、このタイプのすべてのプロバイダー接続のデフォルト設定です。個々のプロバイダーにいずれかのプロパティの設定が含まれている場合、FXOS でその設定が使用され、このデフォルト設定は無視されます。

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis# **scope security**

ステップ 2 セキュリティ RADIUS モードを開始します。

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

ステップ 3 （任意） サーバをダウン状態として通知する前に RADIUS サーバとの通信を再試行する回数を指定します。

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

ステップ 4 （任意）システムがサーバをダウン状態として通知する前に、RADIUS サーバからの応答を待つ時間を設定します。

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

ステップ 5 トランザクションをシステム設定にコミットします。

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

例

次の例は、RADIUS リトライを 4 に設定し、タイムアウト間隔を 30 秒に設定し、トランザクションをコミットします。

```
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 #
```

次のタスク

RADIUS プロバイダーを作成します。

RADIUS プロバイダーの作成

次の手順に従い、RADIUS プロバイダー（このアプライアンスに RADIUS ベースの AAA サービスを提供する特定のリモートサーバー）を定義および設定します。

（注） FXOS では、最大 16 の RADIUS プロバイダーをサポートします。

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis# **scope security**

ステップ 2 セキュリティ RADIUS モードを開始します。

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

ステップ 3 RADIUS サーバ インスタンスを作成し、セキュリティ RADIUS サーバ モードを開始します。

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

ステップ 4 （任意） RADIUS サーバとの通信に使用するポートを指定します。

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

ステップ 5 RADIUS サーバ キーを設定します。

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

キー値を設定するには、**set key** コマンドを入力してから **Enter** を押し、プロンプトでキー値を入力します。

標準ASCII文字を入力できます。ただし、「§」（セクション記号）、「?」（疑問符）、「=」（等号）は除きます。

ステップ 6 （任意） このサーバが試行される順序を指定します。

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

ステップ 7 （任意） サーバをダウン状態として通知する前に RADIUS サーバとの通信を再試行する回数を設定します。

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

ステップ 8 システムがサーバをダウン状態として通知する前に、RADIUS サーバからの応答を待つ時間（秒）を指定します。

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

ヒント RADIUSプロバイダーに二要素認証を選択する場合は、より高い**タイムアウト値**を設定することを推奨します。

ステップ 9 トランザクションをシステム設定にコミットします。

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

例

次の例は、radiusserv7 という名前のサーバインスタンスを作成し、認証ポートを 5858 に設定し、キーを radiuskey321 に設定し、順序を 2 に設定し、再試行回数を 4 回に設定し、タイムアウトを 30 に設定し、トランザクションをコミットします。

```
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 #
```

RADIUS プロバイダーの削除

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis# **scope security**

ステップ 2 セキュリティ RADIUS モードを開始します。

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

ステップ 3 指定したサーバを削除します。

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

ステップ 4 トランザクションをシステム設定にコミットします。

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

例

次の例は、radius1 という RADIUS サーバを削除し、トランザクションをコミットします。

```
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 #
```

TACACS+ プロバイダーの設定

TACACS+ プロバイダーのプロパティの設定

このタスクで設定するプロパティが、このタイプのすべてのプロバイダー接続のデフォルト設定になります。個々のプロバイダーの設定にいずれかのプロパティの設定が含まれている場合、FXOS でその設定が使用され、このデフォルト設定は無視されます。

（注） FXOS シャーシは、Terminal Access Controller Access-Control System Plus（TACACS+）プロトコルのコマンドアカウンティングをサポートしていません。

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis# **scope security**

ステップ 2 セキュリティ TACACS+ モードを開始します。

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

ステップ 3 （任意）システムがサーバをダウン状態として通知する前に、TACACS+ サーバからの応答を待つ時間を設定します。

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

1 ～ 60 秒の整数を入力します。デフォルト値は 5 秒です。

ステップ 4 トランザクションをシステム設定にコミットします。

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

例

次の例は、TACACS+ タイムアウト間隔を 45 秒に設定し、トランザクションをコミットします。

```
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 #
```

次のタスク

TACACS+ プロバイダーを作成します。

TACACS+ プロバイダーの作成

次の手順に従い、TACACS+ プロバイダー（このアプライアンスに TACACS+ ベースの AAA サービスを提供する特定のリモートサーバー）を定義および設定します。

（注） FXOS では、最大 16 の TACACS+ プロバイダーをサポートします。

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis# **scope security**

ステップ 2 セキュリティ TACACS+ モードを開始します。

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

ステップ 3 TACACS+ サーバ インスタンスを作成し、TACACS+ サーバ モードを開始します。

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

ステップ 4 TACACS+ サーバ キーを指定します。

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

キー値を設定するには、**set key** コマンドを入力してから **Enter** を押し、プロンプトでキー値を入力します。

標準ASCII文字を入力できます。ただし、「§」（セクション記号）、「?」（疑問符）、「=」（等号）は除きます。

ステップ 5 （任意） このサーバが試行される順序を指定します。

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

ステップ 6 システムがサーバをダウン状態として通知する前に、TACACS+ サーバからの応答を待つ時間間隔を指定します。

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

ヒント TACACS+ プロバイダーに二要素認証を選択する場合は、より高いタイムアウト値を設定することを推奨します。

ステップ 7 （任意） TACACS+ サーバとの通信に使用するポートを指定します。

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

ステップ 8 トランザクションをシステム設定にコミットします。

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

例

次の例は、tacacsserv680 という名前のサーバインスタンスを作成し、キーを tacacskey321 に設定し、順序を 4 に設定し、認証ポートを 5859 に設定し、トランザクションをコミットします。

```
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 #
```

TACACS+ プロバイダーの削除

手順

ステップ 1 セキュリティ モードを開始します。

Firepower-chassis# **scope security**

ステップ 2 セキュリティ TACACS+ モードを開始します。

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

ステップ 3 指定したサーバを削除します。

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

ステップ 4 トランザクションをシステム設定にコミットします。

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

例

次の例では、tacacs1 という TACACS+ サーバを削除し、トランザクションをコミットします。

```
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 #
```

リモート AAA サーバ設定の確認

ここでは、FXOS CLI を使用して、さまざまなリモート AAA サーバの現行設定を確認する方法について説明します。

現在の FXOS 認証設定の確認

次の例では、 **show authentication** コマンドを使用して現在の FXOS 認証設定を確認する方法を示します。この例では、LDAP が認証のデフォルトモードになります。

```
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 #
```

現在の LDAP 構成の確認

次の例では、ldap モードで **show server detail**コマンドを使用して、現在の LDAP 構成の設定を確認する方法を示します。

```
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-RSA-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 #
```

現在の RADIUS 構成の確認

次の例では、radius モードで **show server detail**コマンドを使用して、現在の RADIUS 構成の設定を確認する方法を示します。

```
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 #
```

現在の TACACS + 設定の確認

次の例では、tacacs モードで **show server detail**コマンドを使用して、現在の TACACS+ 構成の設定を確認する方法を示します。

```
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 #
```

Syslog の設定

システム ロギングは、デバイスから syslog デーモンを実行するサーバへのメッセージを収集する方法です。中央 syslog サーバへロギングは、ログおよびアラートの集約に役立ちます。syslogサービスは、簡単なコンフィギュレーションファイルに従って、メッセージを受信してファイルに保存するか、出力します。この形式のロギングは、ログ用の保護された長期ストレージを提供します。ログは、ルーチンのトラブルシューティングおよびインシデント処理の両方で役立ちます。

手順

ステップ 1 モニタリング モードを開始します。

Firepower-chassis# **scope monitoring**

ステップ 2 コンソールへの syslog の送信を有効化またはディセーブルにします。

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

ステップ 3 （任意）表示するメッセージの最低レベルを選択します。syslogが使用可能である場合、システムはそのレベル以上のメッセージをコンソールに表示します。レベル オプションは緊急性の降順で一覧表示されます。デフォルトのレベルは Critical です。

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

ステップ 4 オペレーティング システムによる syslog 情報のモニタリングを有効化またはディセーブルにします。

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

ステップ 5 （任意）表示するメッセージの最低レベルを選択します。モニタの状態が有効の場合、システムはそのレベル以上のメッセージを表示します。レベル オプションは緊急性の降順で一覧表示されます。デフォルトのレベルは Critical です。

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

（注） **terminal monitor** コマンドを入力した場合にだけ、Critical より下のレベルのメッセージが端末のモニタに表示されます。

ステップ 6 syslog ファイルへの syslog 情報の書き込みを有効化またはディセーブルにします。

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

ステップ 7 メッセージが記録されるファイルの名前を指定します。ファイル名は 16 文字まで入力できます。

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

ステップ 8 （任意）ファイルに保存するメッセージの最低レベルを選択します。ファイルの状態が有効の場合、システムはそのレベル以上のメッセージを syslog ファイルに保存します。レベル オプションは緊急性の降順で一覧表示されます。デフォルトのレベルは Critical です。

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

ステップ 9 （任意）最新のメッセージで最も古いメッセージが上書きされる前の最大ファイルサイズ（バイト単位）を指定します。有効な範囲は 4096 ～ 4194304 バイトです。

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

ステップ 10 最大 3 台の外部 syslog サーバへの syslog メッセージの送信を設定します。

a) 最大 3 台の外部 syslog サーバへの syslog メッセージの送信を有効化またはディセーブルにします。

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

b) （任意）外部ログに保存するメッセージの最低レベルを選択します。リモート宛先が有効になっている場合、システムはそのレベル以上のメッセージを外部サーバに送信します。レベル オプションは緊急性の降順で一覧表示されます。デフォルトのレベルは Critical です。

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

c) 指定したリモート syslog サーバのホスト名または IP アドレスを指定します。ホスト名は 256 文字まで入力できます。

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

d) （任意） 指定したリモート syslog サーバに送信される syslog メッセージに含まれるファシリティ レベルを指定します。

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

ステップ 11 ローカル送信元を設定します。有効化またはディセーブルにするローカル送信元ごとに、次のコマンドを入力します。

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

次のいずれかになります。

- **audits**：すべての監査ログ イベントのロギングを有効または無効にします。
- **events**：すべてのシステム イベントのロギングを有効または無効にします。

• **faults**：すべてのシステム障害のロギングを有効または無効にします。

ステップ 12 トランザクションをコミットします。

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

例

次の例は、ローカル ファイルの syslog メッセージのストレージをイネーブルにし、トランザクションをコミットします。

```
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 #
```

DNS サーバの設定

システムでホスト名の IP アドレスへの解決が必要な場合は、DNS サーバを指定する必要があります。たとえば、DNSサーバーを設定していない場合は、シャーシに関する設定を行うときに、www.cisco.com などの名前を使用できません。サーバの IP アドレスを使用する必要があります。これには、IPv4 または IPv6 アドレスのいずれかを使用できます。最大 4 台の DNS サーバを設定できます。

（注） 複数の DNS サーバを設定する場合、システムによるサーバの検索順はランダムになります。ローカル管理コマンドが DNS サーバの検索を必要とする場合、3 台の DNS サーバのみをランダムに検索します。

手順

ステップ 1 システム モードに入ります。

Firepower-chassis # **scope system**

ステップ 2 システム サービス モードを開始します。

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

ステップ 3 DNS サーバを作成または削除するには、次の該当するコマンドを入力します。

- 指定した IPv4 または IPv6 アドレスの DNS サーバを使用するようにシステムを設定する場合：

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

- 指定した IPv4 または IPv6 アドレスの DNS サーバを削除する場合：

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

ステップ 4 トランザクションをシステム設定にコミットします。

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

例

次の例では、IPv4 アドレス 192.168.200.105 を持つ DNS サーバを設定し、トランザクションをコミットします。

```
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 #
```

次の例では、IPv6 アドレス 2001:db8::22:F376:FF3B:AB3F を持つ DNS サーバを設定し、トランザクションをコミットします。

```
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 #
```

次の例では、IP アドレス 192.168.200.105 を持つ DNS サーバを削除し、トランザクションをコミットします。

```
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 #
```

FIPS モードの有効化

Firepower 4100/9300 シャーシで FIPS モードを有効にするには、次の手順を実行します。

手順

ステップ 1 FXOS CLI から、セキュリティ モードを開始します。

scope security

ステップ 2 FIPS モードを有効にします。

enable fips-mode

ステップ 3 設定を確定します。

commit-buffer

ステップ 4 システムを再起動します。

connect local-mgmt

reboot

FIPS モードが有効になっている場合は、許可されるキーサイズとアルゴリズムが制限されます。MIO は、CiscoSSL と FIPS オブジェクトモジュール（FOM）を使用して暗号化を行います。これにより、ASA 独自の暗号化ライブラリの実装および HW アクセラレーションと比較して、FIPS の検証が容易になります。

次のタスク

FXOS リリース 2.0.1 より以前は、デバイスの最初の設定時に作成した SSH ホスト キーが 1024 ビットにハード コードされていました。FIPS およびコモンクライテリア認定要件に準拠するには、この古いホストキーを破棄し、「SSH ホストキーの生成」で詳細を説明する手順を使用して新しいホストキーを生成する必要があります。これらの追加手順を実行しないと、FIPS モードを有効にしてデバイスをリブートした後に、SSH を使用してスーパバイザに接続できなくなります。FXOS 2.0.1 以降を使用して初期設定を行った場合は、新しいホスト キーを生成する必要はありません。

コモン クライテリア モードの有効化

Firepower 4100/9300 シャーシ上でコモン クライテリア モードを有効にするには、次の手順を実行します。

手順

ステップ 1 FXOS CLI から、セキュリティ モードを開始します。

scope security

ステップ 2 コモン クライテリア モードを有効にします。

enable cc-mode

ステップ 3 設定を確定します。

commit-buffer

ステップ 4 システムを再起動します。

connect local-mgmt

reboot

コモンクライテリア（CC）はコンピュータセキュリティ向け国際基準です。CCは、証明書、監査、ロギング、パスワード、TLS、SSH などに重点を置いています。基本的に FIPS 準拠を前提としています。FIPS と同様に、シスコは、NIST 認定ラボベンダーと契約してテストと NIAP への提出を行っています。

CC モードを有効にすると、サポートする必要があるアルゴリズム、暗号スイート、および機能のリストが制限されます。MIO は、Network Device Collaborative Protection Profile（NDcPP）に対して評価されます。CiscoSSL は、ほとんどがCCコンプライアンスガイドに記載されている要件の一部のみを適用できます。

次のタスク

FXOS リリース 2.0.1 より以前は、デバイスの最初の設定時に作成した SSH ホスト キーが 1024 ビットにハード コードされていました。FIPS およびコモンクライテリア認定要件に準拠するには、この古いホストキーを破棄し、「SSHホストキーの生成」で詳細を説明する手順を使用して新しいホストキーを生成する必要があります。これらの追加手順を実行しないと、コモンクライテリア モードを有効にしてデバイスをリブートした後に、SSH を使用してスーパバイザに接続できなくなります。FXOS 2.0.1 以降を使用して初期設定を行った場合は、新しいホスト キーを生成する必要はありません。

IP アクセスリストの設定

デフォルトでは、Firepower 4100/9300 シャーシはローカル Web サーバへのすべてのアクセスを拒否します。IP アクセスリストを、各 IP ブロックの許可されるサービスのリストを使用して設定する必要があります。

IP アクセスリストは、次のプロトコルをサポートします。

- HTTPS
- SNMP
- SSH

IP アドレス（v4 または v6）の各ブロックで、最大 100 個の異なるサブネットを各サービスに対して設定できます。サブネットを 0、プレフィックスを 0 と指定すると、サービスに無制限にアクセスできるようになります。

手順

ステップ 1 FXOS CLI から、サービス モードを開始します。

scope system

scope services

ステップ 2 アクセスできるようにするサービスの IP ブロックを作成します。

IPv4 の場合

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

IPv6 の場合

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

例

次の例では、IPv4 アドレスブロックを作成、入力、および確認し、SSH にアクセスする方法を示します。

```
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 #
```

次の例では、IPv6 アドレスブロックを作成、入力、および確認し、SSH にアクセスする方法を示します。

```
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 #
```

MAC プール プレフィックスの追加とコンテナ インスタンス インターフェイスの MAC アドレスの表示

FXOS シャーシは、各インスタンスの共有インターフェイスが一意の MAC アドレスを使用するように、コンテナインスタンスインターフェイスのMACアドレスを自動的に生成します。FXOS シャーシは、次の形式を使用して MAC アドレスを生成します。

A2*xx.yyzz.zzzz*

xx.yy はユーザ定義のプレフィックスまたはシステム定義のプレフィックスであり、*zz.zzzz* はシャーシが生成した内部カウンタです。システム定義のプレフィックスは、IDPROMにプログラムされている Burned-in MAC アドレス内の最初の MAC アドレスの下部 2 バイトと一致します。**connect fxos**を使用し、次に**show module**を使用して、MACアドレスプールを表示します。たとえば、モジュール 1 について示されている MAC アドレスの範囲が b0aa.772f.f0b0 ～ b0aa.772f.f0bf の場合、システム プレフィックスは f0b0 になります。

詳細については、「コンテナ インスタンス インターフェイスの自動 MAC アドレス」を参照してください。

この手順では、MAC アドレスの表示方法と生成で使用されるプレフィックスのオプションの定義方法について説明します。

（注） 論理デバイスの展開後に MAC アドレスのプレフィックスを変更すると、トラフィックが中断される可能性があります。

手順

ステップ 1 セキュリティ サービス モードを開始してから、自動 MAC プール モードを開始します。

scope ssa

scope auto-macpool

例：

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

ステップ 2 MAC アドレスの生成時に使用される MAC アドレスのプレフィックスを設定します。

set prefix *prefix*

- *prefix*：1 ～ 65535 の 10 進数を入力します。このプレフィックスは 4 桁の 16 進数値に変換され、MAC アドレスの一部として使用されます。

プレフィックスの使用方法を示す例の場合、プレフィックス 77 を設定すると、シャーシは 77 を 16 進数値 004D（*yyxx*）に変換します。MAC アドレスで使用すると、プレフィックスはシャーシ ネイティブ形式に一致するように逆にされます（*xxyy*）。

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

プレフィックス 1009（03F1）の場合、MAC アドレスは次のようになります。

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

例：

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

ステップ 3 設定を保存します。

commit-buffer

例：

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

ステップ 4 MAC アドレスの割り当てを表示します。

show mac-address

例：

```
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
```

例

次の例では、MAC プレフィックスを 33 に設定しています。

```
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 #
```

コンテナインスタンスにリソースプロファイルを追加

コンテナインスタンスごとにリソース使用率を指定するには、1つまたは複数のリソースプロファイルを作成します。論理デバイス/アプリケーション インスタンスを展開するときに、使用するリソース プロファイルを指定します。リソース プロファイルは CPU コアの数を設定します。RAM はコアの数に従って動的に割り当てられ、ディスク容量はインスタンスごとに 40 GB に設定されます。

- コアの最小数は 6 です。

（注） コア数が少ないインスタンスは、コア数が多いインスタンスよりも、CPU 使用率が比較的高くなる場合があります。コア数が少ないインスタンスは、トラフィック負荷の変化の影響を受けやすくなります。トラフィックのドロップが発生した場合には、より多くのコアを割り当ててください。

- コアは偶数（6、8、10、12、14 など）で最大値まで割り当てることができます。
- 利用可能な最大コア数は、セキュリティモジュール/シャーシモデルによって異なります。「コンテナインスタンスの要件と前提条件」を参照してください。

シャーシには、「Default-Small」と呼ばれるデフォルトリソースプロファイルが含まれています。このコア数は最小です。このプロファイルの定義を変更したり、使用されていない場合には削除することもできます。シャーシをリロードし、システムに他のプロファイルが存在しない場合は、このプロファイルが作成されます。

使用中のリソースプロファイルの設定を変更することはできません。そのリソースプロファイルを使用しているすべてのインスタンスを無効にしてから、リソースプロファイルを変更し、最後にインスタンスを再度有効にする必要があります。確立されたハイ アベイラビリティ ペアまたはクラスタ内のインスタンスのサイズを変更する場合、できるだけ早くすべてのメンバを同じサイズにする必要があります。

Firepower Threat Defense インスタンスを FMC に追加した後にリソースプロファイルの設定を変更する場合は、FMC の [デバイス（Devices）] > [デバイス管理（Device Management）] > [デ

バイス（Device）] > [システム（System）] > [インベントリ（Inventory）] ダイアログボックスで各ユニットのインベントリを更新します。

手順

ステップ 1 セキュリティ サービス モードを開始します。

scope ssa

例：

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

ステップ 2 リソースプロファイルを作成します。

enter resource-profile *name*

- [name]：プロファイルの名前を 1 ～ 64 文字で設定します。追加後にこのプロファイルの名前を変更することはできません。

例：

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

ステップ 3 説明を入力します。

set description *description*

- [description]：プロファイルの説明を最大 510 文字で設定します。フレーズを引用符（"）で囲み、スペースを追加します。

例：

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

ステップ 4 CPU コア数を設定します。

set cpu-core-count *cores*

- [cores]：プロファイルのコア数を 6 ～ 最大数（偶数）で設定します。最大数はシャーシによって異なります。

例：

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

ステップ 5 設定を保存します。

commit-buffer

例：

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

ステップ 6 セキュリティサービスモードからリソースプロファイルの割り当てを表示します。

show resource-profile user-defined

例：

```
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
```

ステップ 7 セキュリティ モジュール/エンジン スロットのリソース使用率を表示します。

show monitor detail

例：

```
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
```

ステップ 8 アプリケーション インスタンスのリソース割り当てを表示します。

show resource detail

例：

```
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
```

例

次の例では、3 つのリソース プロファイルを追加します。

```
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 #
```

ネットワーク制御ポリシーの設定

他社製デバイスのディスカバリを許可するために、FXOS は、IEEE 802.1ab 規格で定義されているベンダーニュートラルなデバイス ディスカバリ プロトコルである *Link Layer Discovery Protocol*（*LLDP*）をサポートしています。LLDP を使用すると、ネットワークデバイスはネットワーデバイスに関する情報を、ネットワーク上の他のデバイスにアドバタイズできます。このプロトコルはデータリンク層で動作するため、異なるネットワーク層プロトコルが稼働する 2 つのシステムで互いの情報を学習できます。

LLDP は、デバイスおよびそのインターフェイスの機能と現在のステータスに関する情報を送信する単一方向のプロトコルです。LLDP デバイスはこのプロトコルを使用して、他の LLDP デバイスからだけ情報を要求します。

To enable this functionality on your FXOS chassis, you can configure a network control policy, which specifies LLDP transmission and receiving behavior. ネットワーク制御ポリシーを作成した後、インターフェイスに割り当てる必要があります。固定ポート、EPM ポート、ポートチャネル、およびブレイクアウトポートなどの任意の前面インターフェイスでLLDPを有効にできます。

（注）
- LLDP is not configurable on dedicated management ports.
- ブレードに接続する内部バックプレーンポートではデフォルトでLLDPが有効になっています。無効にするオプションはありません。他のすべてのポートでは、LLDP はデフォルトで無効になっています。

手順

ステップ 1 組織の範囲を入力します。

scope org

例：

```
Firepower # scope org
```

ステップ 2 Create and enable the network control policy.

create nw-ctrl-policy *nw-policy*

例：

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

ステップ 3 LLDP をイネーブルにします。

enable lldp *{receive | transmit}*

例：

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

ステップ 4 設定をコミットします。

commit-buffer

例：

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

ステップ 5 Specify whether to enable or disable LLDP for receiving/transmitting.

enable lldp *receive/transmit*

commit-buffer

例：

```
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
```

ステップ 6 次のコマンドを使用して、ネットワーク制御ポリシーをインターフェイスに適用します。

a) インターフェイスを入力します。

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) 変更内容を表示します。

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) 設定をコミットします。

commit-buffer

例：

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

シャーシ URL の設定

管理 URL を指定して、FMC から直接、 Firepower Threat Defense インスタンスの Firepower Chassis Manager を簡単に開くことができます。シャーシ管理 URL を指定しない場合には、代わりにシャーシ名が使用されます。

Firepower Threat Defense インスタンスを FMC に追加した後にシャーシ URL 設定を変更する場合は、**[Devices]** > **[Device Management]** > **[Device]** > **[System]** > **[Inventory]** ダイアログボックスで各ユニットのインベントリを更新します。

手順

ステップ 1 システム モードに入ります。

scope system

例：

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

ステップ 2 新しいシャーシ名を設定するには、次のコマンドを実行します。

set name *chassis_name*

- *chassis_name*：シャーシの名前を 1 ～ 60 文字で設定します。

例：

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

ステップ 3 管理 URL を設定するには、次のコマンドを実行します。

set mgmt-url *management_url*

- *management_url*：Firepower Chassis Manager 内で FMC が Firepower Threat Defense インスタンスに接続するために使用する URL を設定します。URL は https:// で始まる必要があります。シャーシ管理 URL を指定しない場合、代わりにシャーシ名が使用されます。

例：

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

ステップ 4 設定を保存します。

commit-buffer

例：

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

ステップ 5 設定を表示します。

show detail

例：

```
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
```

脆弱キー交換アルゴリズムの変更

機器で使用する脆弱キー交換アルゴリズムは、次の方法で緩和できます。

- FIPS/CC モードの設定
- 暗号スイートの設定

FIPS/CC モードの設定

手順

ステップ 1 FXOS CLI から、セキュリティ モードを開始します。

scope security

ステップ 2 FIPS モードを有効にします。

enable fips-mode

ステップ 3 設定をコミットします。
commit-buffer

暗号スイートの設定

手順

ステップ 1 システム モードに入ります。
Firepower-chassis# **scope system**

ステップ 2 システム サービス モードを開始します。
Firepower-chassis /system # **scope services**

ステップ 3 HTTPS サービスを表示します。
Firepower-chassis /system/services # **show https**

ステップ 4 暗号スイート モードを設定します。
Firepower-chassis /system/services # **set https cipher-suite-mode** *custom*

ステップ 5 暗号スイート文字列を設定する
Firepower-chassis /system/services # set https cipher-suite *******

ステップ 6 設定をシステム構成に対して確定します。
Firepower-chassis /system/services # **commit-buffer**

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。