# UCS Manager コミュニケーションサービス

この章は、次の項で構成されています。



## 通信サービス

以下に定義する通信サービスを使用してサードパーティアプリケーションを Cisco UCS に接続できます。

Cisco UCS Manager では、次のサービスに対して IPv4 および IPv6 アドレス アクセスをサポートしています。

- CIM XML
- HTTP
- HTTPS
- SNMP
- SSH
- Telnet

Cisco UCS Manager では、Web ブラウザから [Cisco UCS KVM Direct] 起動ページへのアウトオブバンド IPv4 アドレス アクセスをサポートしています。このアクセスを提供するには、次のサービスをイネーブルにする必要があります。

- CIMC Web サービス

| 通信サービス | 説明 |
|---|---|
| CIM XML | Common Information Model（CIM XML）サービスはデフォルトはディセーブルであり、読み取り専用モードでのみ利用できます。デフォルトのポートは 5988 です。<br>CIM XML は、Distributed Management Task Force によって定義された CIM 情報を交換するための標準ベースのプロトコルです。 |
| CIMC Web サービス | このサービスは、デフォルトでディセーブルになります。<br>このサービスをイネーブルにすると、ユーザは直接サーバに割り当てられるか、またはサービス プロファイルを介しサーバに関連付けられたアウトオブバンドの管理 IP アドレスの 1 つを使用して直接サーバ CIMC にアクセスできます。<br>（注） CIMC Web サービスは全体的にイネーブルまたはディセーブルにすることのみが可能です。個別の CIMC IP アドレスに対し KVM ダイレクト アクセスを設定できません。 |
| HTTP | デフォルトでは、HTTP はポート 80 でイネーブルになっています。<br>Cisco UCS Manager GUI は HTTP または HTTPS のブラウザで実行できます。HTTP を選択した場合、すべてのデータはクリア テキスト モードで交換されます。<br>ブラウザ セッションの安全性の理由により、HTTPS をイネーブルにし、HTTP をディセーブルにすることを推奨します。<br>デフォルトでは、Cisco UCS では同等の HTTPS にリダイレクトするブラウザ リダイレクトを実装しています。この動作は変更しないことを推奨します。<br>（注） Cisco UCS バージョン 1.4(1) にアップグレードすると、セキュアなブラウザへのブラウザのリダイレクトはデフォルトでは発生しなくなります。HTTP ブラウザからの同等の HTTPS ブラウザへリダイレクトするには、Cisco UCS Manager で [Redirect HTTP to HTTPS] をイネーブルにします。 |
| HTTPS | デフォルトでは、HTTPS はポートでイネーブルになっています。<br>HTTPS を使用すると、すべてのデータはセキュアなサーバを介して暗号化モードで交換されます。<br>ブラウザ セッションの安全性の理由により、HTTPS だけを使用し、HTTP 通信はディセーブルにするかリダイレクトすることを推奨します。 |

| 通信サービス | 説明 |
|---|---|
| SMASH CLP | このサービスは読み取り専用アクセスに対してイネーブルになり、show コマンドなど、プロトコルの一部のサブセットをサポートします。これをディセーブルにすることはできません。<br>このシェル サービスは、Distributed Management Task Force によって定義された標準の 1 つです。 |
| SNMP | デフォルトでは、このサービスはディセーブルになっています。イネーブルの場合、デフォルトのポートは161です。コミュニティと少なくとも 1 つの SNMP トラップを設定する必要があります。<br>システムに SNMP サーバとの統合が含まれる場合にだけこのサービスをイネーブルにします。 |
| SSH | このサービスは、ポート22でイネーブルになります。これはディセーブルにできず、デフォルトのポートを変更することもできません。<br>このサービスは Cisco UCS Manager CLI へのアクセスを提供します。 |
| Telnet | デフォルトでは、このサービスはディセーブルになっています。<br>このサービスは Cisco UCS Manager CLI へのアクセスを提供します。 |

# 非セキュアなコミュニケーション サービス

## Web セッション制限の設定

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope system** . | システム モードを開始します。 |
| ステップ 2 | UCS-A /system # **scope services** . | サービス モードを開始します。 |
| ステップ 3 | UCS-A /system/services # **scope web-session-limits** . | Web セッションの制限モードを開始します。 |
| ステップ 4 | UCS-A /system/services/web-session-limits # **set** {**maximum-event-interval** \|**per-user** \| **total**}*number* . | 次の Web セッション制限を設定できます。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| | | **名前** / **説明**<br>**Maximum Sessions Per User** / 各ユーザに許可される HTTP および HTTPS の同時セッションの最大数。<br>1 ～ 256 の整数を入力します。<br>**Maximum Sessions** / システム内のすべてのユーザに許可される HTTP および HTTPS の同時セッションの最大数。<br>1 ～ 256 の整数を入力します。<br>**[Maximum Event Interval (in seconds)]** / 2 つのイベント間の最大時間間隔。UI からのユーザ要求に対する応答など、さまざまなタイプのイベント変更通知を追跡します。時間間隔が経過すると、UI セッションは終了します。<br>120 ～ 3600 の整数を入力します。 |
| **ステップ 5** | UCS-A /system/services/web-session-limits # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

**例**

次に、最大イベント間隔を設定する方法を示します。

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # scope web-session-limits
UCS-A /system/services/web-session-limits # set maximum-event-interval 300
UCS-A /system/services/web-session-limits # commit buffer
```

# Web セッション制限の表示

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope system** | システム モードを開始します。 |
| ステップ 2 | UCS-A /system # **scope services** | サービス モードを開始します。 |
| ステップ 3 | /system/services # **show web-session-limits** | Web セッションの設定を表示します。 |

例

次の例では、Web セッションの制限を表示する方法を示します。

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # show web-session-limits
Web Sessions:
    Maximum logins for single user Maximum Sessions Maximum Event Interval (sec)
    ------------------------------ ---------------- ----------------------------
    32                             256              600
UCS-A /system/services #
```

# シェル セッション制限の設定

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope system** . | システム モードを開始します。 |
| ステップ 2 | UCS-A /system # **scope services** . | サービス モードを開始します。 |
| ステップ 3 | UCS-A /system/services # **scope shell-session-limits** | |
| ステップ 4 | UCS-A /system/services/shell-session-limits # **set** {**per-user** \|**total**}番号 | 次のシェル セッション制限を設定できます。<br>名前：**Maximum Sessions Per User**<br>説明：ユーザごとに許可される同時シェル セッションの最大数。<br>1 ～ 32 の整数を入力します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| | | 名前 / 説明<br>**Maximum Sessions** / システム内のすべてのユーザに許可される同時シェルセッションの最大数。<br>1 ～ 32 の整数を入力します。 |
| ステップ 5 | UCS-A /system/services/shell-session-limits # **commit-buffer** . | トランザクションをシステムの設定にコミットします。 |

例

次に、最大セッション数を設定する例を示します。

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # scope shell-session-limits
UCS-A /system/services/shell-session-limits # set maximum-sessions 20
UCS-A /system/services/shell-session-limits # commit buffer
```

# Viewing Shell Session Limits

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope system** | システム モードを開始します。 |
| ステップ 2 | UCS-A /system # **scope services** | サービス モードを開始します。 |
| ステップ 3 | /system/services # **show shell-session-limits** | シェルセッションの設定を表示します。 |

例

次の例では、シェル セッションの制限を表示する方法を示します。

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # show shell-session-limits
Shell Sessions:
    Maximum logins for single user Maximum Sessions
    ------------------------------ ----------------
    32                             32
UCS-A /system/services #
```

# CIM XML の設定

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope system** | システム モードを開始します。 |
| ステップ 2 | UCS-A /system # **scope services** | システム サービス モードを開始します。 |
| ステップ 3 | UCS-A /system/services # **enable cimxml** | CIM XLM サービスを有効にします。 |
| ステップ 4 | UCS-A /system/services # **set cimxml port** *port-num* | CIM XML 接続のポートを指定します。 |
| ステップ 5 | UCS-A /system/services # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

例

次に、CIM XML を有効にし、ポート番号を 5988 に設定し、トランザクションをコミットする例を示します。

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # enable cimxml
UCS-A /system/services* # set cimxml port 5988
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

# HTTP の設定

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope system** | システム モードを開始します。 |
| ステップ 2 | UCS-A /system # **scope services** | システム サービス モードを開始します。 |
| ステップ 3 | UCS-A /system/services # **enable http** | HTTP サービスを有効にします。 |
| ステップ 4 | UCS-A /system/services # **set http port** *port-num* | HTTP 接続で使用されるポートを指定します。 |
| ステップ 5 | UCS-A /system/services # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

### 例

次の例は、HTTP を有効にし、ポート番号を 80 に設定し、トランザクションをコミットします。

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # enable http
UCS-A /system/services* # set http port 80
Warning: When committed, this closes all the web sessions.
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

## HTTP の設定解除

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope system** | システム モードを開始します。 |
| **ステップ 2** | UCS-A /system # **scope services** | システム サービス モードを開始します。 |
| **ステップ 3** | UCS-A /system/services # **disable http** | HTTP サービスを無効にします。 |
| **ステップ 4** | UCS-A /system/services # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

### 例

次に、HTTP 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。

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # disable http
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

# セキュアなコミュニケーション サービス

## 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

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope system** | システム モードを開始します。 |
| **ステップ 2** | UCS-A /system # **scope services** | システム サービス モードを開始します。 |
| **ステップ 3** | UCS-A /system/services # **enable https** | HTTPS サービスを有効にします。 |
| **ステップ 4** | （任意） UCS-A /system/services # **set https port** *port-num* | HTTPS 接続で使用されるポートを指定します。 |
| **ステップ 5** | （任意） UCS-A /system/services # **set https keyring** *keyring-name* | HTTPS に対して作成したキー リングの名前を指定します。 |
| **ステップ 6** | （任意） UCS-A /system/services # **set https cipher-suite-mode** *cipher-suite-mode* | Cisco UCS 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。<br>*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。<br>• **high-strength**<br>• **medium-strength**<br>• **low-strength**<br>• **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） UCS-A /system/services # **set https cipher-suite** *cipher-suite-spec-string* | **cipher-suite-mode** が **custom** に設定されている場合、この Cisco UCS ドメインに対する暗号スイート セキュリティのカスタム レベルを指定します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| | | *cipher-suite-spec-string* 最大 256 文字まで使用できますが、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。.<br>たとえば、Cisco UCS Manager 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。<br>**ALL:!ADH:!EXPORT56:!LOW:RC4+RSA:+HIGH:+MEDIUM:+EXP:+eNULL**<br>（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） UCS-A /system/services # **set https ssl-protocol** | UCSM が許可する SSL プロトコルを選択できます。値は [Default (Allow all except SSLv2 and SSLv3)] と [Only TLSV1.2] です。[Only TLSv1.2] を選択すると、低いバージョンの TLS プロトコルを使用した Web クライアントからの接続は確立されません。 |
| **ステップ 9** | UCS-A /system/services # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

### 例

次の例では、HTTPS を有効にし、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、Web サーバを TLSv1.2 を使用した接続のみを受け付けるように設定し、トランザクションをコミットします。

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # enable https
UCS-A /system/services* # set https port 443
Warning: When committed, this closes all the web sessions.
UCS-A /system/services* # set https keyring kring7984
UCS-A /system/services* # set https cipher-suite-mode high
UCS-A /system/services* # set https ssl-protocol tls1-2
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

# HTTPS の設定解除

### 始める前に

HTTP から HTTPS へのリダイレクションを無効にします。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope system** | システム モードを開始します。 |
| **ステップ 2** | UCS-A /system # **scope services** | システム サービス モードを開始します。 |
| **ステップ 3** | UCS-A /system/services # **disable https** | HTTPS サービスを無効にします。 |
| **ステップ 4** | UCS-A /system/services # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

### 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。

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # disable https
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

# 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 Cisco UCS Manager 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。Cisco UCS Manager では最初に 1024 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、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。

この操作は、UCS Manager CLI のみで使用できます。

### 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、Cisco UCS Manager 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。

UCS M5、M4、および M3サーバの CIMC の自己署名 KVM 証明書を、ユーザが生成したパブリック証明書に変更できます。ただし、パスワードで保護された X.509 証明書秘密キーはサポートされません。KVM 証明書の作成 （18 ページ）このプロセスに関する詳細情報を提供します。

**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。

### トラスト ポイント

Cisco UCS Manager 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、Cisco UCS Manager 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。

## 信頼できない CA 署名付き証明書の作成

パブリック認証局（CA）を使用して証明書の生成と署名を行う代わりに、独自の CA を運用して独自の証明書に署名することができます。証明書とキーのペアを生成するには、2048 ビットの RSA鍵と x.509 PEM 証明書を生成する必要があります。このセクションでは、Linux で実行されている OpenSSL 証明書サーバを使用して CA を作成するコマンドおよび証明書を生成するコマンドについて説明します。OpenSSL の詳細については、http://www.openssl.org を参照してください。

（注） これらのコマンドは、OpenSSL パッケージを使用している Linux サーバで入力します。

### 始める前に

組織内のサーバで、証明書サーバのソフトウェア パッケージを取得してインストールします。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **openssl genrsa -out** *CA_keyfilename keysize*<br>例：<br>`# openssl genrsa -out cert.private 2048` | このコマンドは、CA で使用される RSA 秘密キーを生成します。<br>指定されたファイル名には、指定されたサイズの RSA キーが含まれています。 |
| ステップ 2 | **openssl req -new -x509 -days** *numdays* **-key** *CA_keyfilename* **-out** *CA_certfilename*<br>例：<br>`# openssl req -new -x509 -days 365 -key cert.private -out cert.pem` | このコマンドは、指定されたキーを使用して、CA の自己署名証明書を新規に作成します。証明書は指定された期間有効になります。このコマンドは、ユーザに証明書の追加情報を求めるプロンプトを表示します。<br>証明書サーバは、アクティブな CA です。 |
| ステップ 3 | （任意） **openssl x509 -req -days** *numdays* **-in** *CSR_filename* **-CA** *CA_certfilename* **-set_serial 04 -CAkey** *CA_keyfilename* **-out** *server_certfilename* **-extfile openssl.conf**<br>例：<br>`# openssl x509 -req -days 365 -in csr.txt -CA cert.pem -set_serial 04 -CAkey cert.private -out myserver05.crt -extfile openssl.conf` | このコマンドは、CA が CSR ファイルを使用してサーバ証明書を生成するように指示します。<br>サーバ証明書は、出力ファイルに含まれています。 |

### 例

この例は、CA の作成方法、および新規に作成された CA が署名するサーバ証明書の生成方法を示します。これらのコマンドは、OpenSSL を実行している Linux サーバで入力します。

```
# /usr/bin/openssl genrsa -out cert.private 2048 Generating RSA private key,
2048 bit long modulus .............++++++ .....++++++ e is 65537 (0x10001) #
/usr/bin/openssl req -new -x509 -days 365 -key cert.private -out cert.pem You
 are about to be asked to enter information that will be incorporated into your
 certificate request. What you are about to enter is what is called a
Distinguished Name or a DN. There are quite a few fields but you can leave some
 blank For some fields there will be a default value, If you enter '.', the
field will be left blank. ----- Country Name (2 letter code) [GB]:US State or
 Province Name (full name) [Berkshire]:California Locality Name (eg, city)
[Newbury]:San Jose Organization Name (eg, company) [My Company Ltd]:Example
Incorporated Organizational Unit Name (eg, section) []:Unit A Common Name (eg,
```

```
 your name or your server's hostname) []:example.com Email Address
[]:admin@example.com # /usr/bin/openssl x509 -req -days 365 -in csr.txt -CA
cert.pem -set_serial 01 -CAkey cert.private -out server.crt -extfile openssl.conf
 Signature ok subject=/C=US/ST=California/L=San Jose/O=Example Inc./OU=Unit
A/CN=example.com/emailAddress=john@example.com Getting CA Private Key #
```

# キー リングの作成

Cisco UCS Manager 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

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope security** | セキュリティ モードを開始します。 |
| **ステップ 2** | UCS-A /security # **create keyring** *keyring-name* | キー リングを作成し、名前を指定します。 |
| **ステップ 3** | UCS-A /security/keyring # **set modulus** {**mod1024** \| **mod1536** \| **mod2048** \| **mod512**} | SSL キーのビット長を設定します。 |
| **ステップ 4** | UCS-A /security/keyring # **commit-buffer** | トランザクションをコミットします。 |

### 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。

```
UCS-A# scope security
UCS-A /security # create keyring kr220
UCS-A /security/keyring* # set modulus mod1024
UCS-A /security/keyring* # commit-buffer
UCS-A /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、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。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope security** | セキュリティ モードを開始します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 2 | UCS-A /security # **scope keyring default** | デフォルト キー リングでキー リング セキュリティ モードを開始します。 |
| ステップ 3 | UCS-A /security/keyring # **set regenerate yes** | デフォルト キー リングを再生成します。 |
| ステップ 4 | UCS-A /security/keyring # **commit-buffer** | トランザクションをコミットします。 |

### 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。

```
UCS-A# scope security
UCS-A /security # scope keyring default
UCS-A /security/keyring* # set regenerate yes
UCS-A /security/keyring* # commit-buffer
UCS-A /security/keyring #
```

# 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

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope security** | セキュリティ モードを開始します。 |
| ステップ 2 | UCS-A /security # **scope keyring** *keyring-name* | キー リングのコンフィギュレーション モードを開始します。 |
| ステップ 3 | UCS-A /security/keyring # **create certreq** {**ip** [*ipv4-addr* \| *ipv6-v6*] \|**subject-name** *name*} | 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。 |
| ステップ 4 | UCS-A /security/keyring/certreq # **commit-buffer** | トランザクションをコミットします。 |
| ステップ 5 | UCS-A /security/keyring # **show certreq** | 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。 |

### 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。

```
UCS-A# scope security
UCS-A /security # scope keyring kr220
UCS-A /security/keyring # create certreq ip 192.168.200.123 subject-name sjc04
Certificate request password:
Confirm certificate request password:
UCS-A /security/keyring* # commit-buffer
UCS-A /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-----

UCS-A /security/keyring #
```

# 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

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope security** | セキュリティ モードを開始します。 |
| **ステップ 2** | UCS-A /security # **scope keyring** *keyring-name* | キー リングのコンフィギュレーション モードを開始します。 |
| **ステップ 3** | UCS-A /security/keyring # **create certreq** | 証明書要求を作成します。 |
| **ステップ 4** | UCS-A /security/keyring/certreq* # **set country** *country name* | 会社が存在している国の国コードを指定します。 |
| **ステップ 5** | UCS-A /security/keyring/certreq* # **set dns** *DNS Name* | 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。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 6 | UCS-A /security/keyring/certreq* # **set e-mail** *E-mail name* | 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。 |
| ステップ 7 | UCS-A /security/keyring/certreq* # **set ip** *certificate request ip-address* \| **ipv6** *certificate request ipv6-address* | Cisco UCS ドメインの IPv4 または IPv6 アドレスを指定します。 |
| ステップ 8 | UCS-A /security/keyring/certreq* # **set fi-a-ip** *certificate request FI A ip-address* \| **fi-a-ipv6** *certificate request FI A ipv6-address* | ファブリック インターコネクト A の IPv4 または IPv6 アドレスを指定します。 |
| ステップ 9 | UCS-A /security/keyring/certreq* # **set fi-b-ip** *certificate request FI B ip-address* \| **fi-b-ipv6** *certificate request FI B ipv6-address* | ファブリック インターコネクト B の IPv4 または IPv6 アドレスを指定します。 |
| ステップ 10 | UCS-A /security/keyring/certreq* # **set locality** *locality name (eg, city)* | 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。 |
| ステップ 11 | UCS-A /security/keyring/certreq* # **set org-name** *organization name* | 証明書を要求している組織を指定します。 |
| ステップ 12 | UCS-A /security/keyring/certreq* # **set org-unit-name** *organizational unit name* | 組織ユニットを指定します。 |
| ステップ 13 | UCS-A /security/keyring/certreq* # **set password** *certificate request password* | 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。 |
| ステップ 14 | UCS-A /security/keyring/certreq* # **set state** *state, province or county* | 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。 |
| ステップ 15 | UCS-A /security/keyring/certreq* # **set subject-name** *certificate request name* | ファブリック インターコネクトの完全修飾ドメイン名を指定します。 |
| ステップ 16 | UCS-A /security/keyring/certreq* # **commit-buffer** | トランザクションをコミットします。 |
| ステップ 17 | UCS-A /security/keyring # **show certreq** | 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。 |

### 例

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。

```
UCS-A# scope security
UCS-A /security # scope keyring kr220
```

```
UCS-A /security/keyring # create certreq
UCS-A /security/keyring/certreq* # set ip 192.168.200.123
UCS-A /security/keyring/certreq* # set fi-a-ip 192.168.200.124
UCS-A /security/keyring/certreq* # set fi-b-ip 192.168.200.125
UCS-A /security/keyring/certreq* # set subject-name sjc04
UCS-A /security/keyring/certreq* # set country US
UCS-A /security/keyring/certreq* # set dns bgl-samc-15A
UCS-A /security/keyring/certreq* # set e-mail test@cisco.com
UCS-A /security/keyring/certreq* # set locality new york city
UCS-A /security/keyring/certreq* # set org-name "Cisco Systems"
UCS-A /security/keyring/certreq* # set org-unit-name Testing
UCS-A /security/keyring/certreq* # set state new york
UCS-A /security/keyring/certreq* # commit-buffer
UCS-A /security/keyring/certreq # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request FI A ip address: 192.168.200.124
Certificate request FI B ip address: 192.168.200.125
Certificate request e-mail name: test@cisco.com
Certificate request ipv6 address: ::
Certificate request FI A ipv6 address: ::
Certificate request FI B ipv6 address: ::
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-----

UCS-A /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。

# KVM 証明書の作成

この手順を使用して、KVM 証明書を作成できます。この操作により、CIMC がリブートします。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope server** *chassis-id / blade-id* | 指定サーバのシャーシ サーバ モードを開始します。 |
| **ステップ 2** | UCS-A /chassis/server # **scope cimc** | シャーシ サーバ CIMC モードを開始します。 |
| **ステップ 3** | UCS-A /chassis/server/cimc # **create kvm-certificate** | KVM 証明書を作成します。 |
| **ステップ 4** | UCS-A /chassis/server/cimc/kvm-certificate* # set certificate | 指定されたユーザ生成のパブリック証明書を設定します。 |
| **ステップ 5** | UCS-A /chassis/server/cimc/kvm-certificate* # set key | 対応するユーザ生成の秘密キーを設定します。<br>（注） パスワード保護された X.509 証明書の秘密キーはサポートされていません。 |
| **ステップ 6** | UCS-A /chassis/server/cimc/kvm-certificate* # **commit-buffer** | トランザクションをシステムの設定にコミットします。<br>この操作により、CIMC がリブートします。 |

### 例

次に、KVM 証明書を作成し、トランザクションをコミットする例を示します。

```
UCS-A# scope server 1/3
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # create kvm-certificate
UCS-A /chassis/server/cimc/kvm-certificate* # set certificate
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Prompt Certificate:
>
...

UCS-A /chassis/server/cimc/kvm-certificate* # set key
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Prompt Key:
>
...

UCS-A /chassis/server/cimc/kvm-certificate* # commit-buffer
UCS-A /chassis/server/cimc/kvm-certificate #
```

# KVM 証明書のクリア

この操作により、CIMC がリブートします。

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope server** *chassis-id* / *blade-id* | 指定サーバのシャーシ サーバ モードを開始します。 |
| ステップ 2 | UCS-A /chassis/server # **scope cimc** | シャーシ サーバ CIMC モードを開始します。 |
| ステップ 3 | UCS-A /chassis/server/cimc # **clear kvm-certificate** | KVM 証明書をクリアします。 |
| ステップ 4 | UCS-A /chassis/server/cimc* # **commit-buffer** | トランザクションをシステムの設定にコミットします。<br>この操作により、CIMC がリブートします。 |

例

次に、KVM 証明書をクリアし、トランザクションをコミットする例を示します。

```
UCS-A# scope server 1/3
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # clear kvm-certificate
Warning: When committed, this operation will result in CIMC reboot.
UCS-A /chassis/server/cimc* # commit-buffer
UCS-A /chassis/server/cimc #
```

# トラスト ポイントの作成

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope security** | セキュリティ モードを開始します。 |
| ステップ 2 | UCS-A /security # **create trustpoint** *name* | トラスト ポイントを作成し、その名前を指定します。 |
| ステップ 3 | UCS-A /security/trustpoint # **set certchain** [*certchain*] | このトラスト ポイントの証明書情報を指定します。<br>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 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| | | 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。<br>**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** | UCS-A /security/trustpoint # **commit-buffer** | トランザクションをコミットします。 |

### 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。

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

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope security** | セキュリティ モードを開始します。 |
| **ステップ 2** | UCS-A /security # **scope keyring** *keyring-name* | 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 モードを開始します。 |
| **ステップ 3** | UCS-A /security/keyring # **set trustpoint** *name* | 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。 |
| **ステップ 4** | UCS-A /security/keyring # **set cert** | 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。<br>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。<br>**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** | UCS-A /security/keyring # **commit-buffer** | トランザクションをコミットします。 |

### 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。

```
UCS-A# scope security
UCS-A /security # scope keyring kr220
UCS-A /security/keyring # set trustpoint tPoint10
UCS-A /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/SINv7wd6Vv2pBt6ZpXgD4VBNKOND1
> GMbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bF5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzCl903O6Mg51zq1zXcz75+VFj2I6rH9asckCld3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtvlWvfhevskV0j6
> mK3Ku+YiORnv6DhxrOoqau8r/hyI/L43l7IPN1HhOi3oha4=
> -----END CERTIFICATE-----
> ENDOFBUF
UCS-A /security/keyring* # commit-buffer
UCS-A /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。

# キーリングの削除

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope security** | セキュリティ モードを開始します。 |
| **ステップ 2** | UCS-A /security # **delete keyring** *name* | 名前付きのキー リングを削除します。 |
| **ステップ 3** | UCS-A /security # **commit-buffer** | トランザクションをコミットします。 |

### 例

次の例では、キー リングを削除します。

```
UCS-A# scope security
UCS-A /security # delete keyring key10
UCS-A /security* # commit-buffer
UCS-A /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

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope security** | セキュリティ モードを開始します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 2 | UCS-A /security # **delete trustpoint** *name* | 指定したトラスト ポイントを削除します。 |
| ステップ 3 | UCS-A /security # **commit-buffer** | トランザクションをコミットします。 |

### 例

次に、トラスト ポイントを削除する例を示します。

```
UCS-A# scope security
UCS-A /security # delete trustpoint tPoint10
UCS-A /security* # commit-buffer
UCS-A /security #
```

# HTTPS への HTTP リダイレクションの有効化

### 始める前に

HTTP と HTTPS の両方を有効にします。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope system** | システム モードを開始します。 |
| ステップ 2 | UCS-A /system # **scope services** | システム サービス モードを開始します。 |
| ステップ 3 | UCS-A /system/services # **enable http-redirect** | HTTP リダイレクト サービスを有効にします。<br>イネーブルの場合、HTTP 経由で試行される通信はすべて同等の HTTPS アドレスにリダイレクトされます。<br>このオプションは、この Cisco UCS ドメインへの HTTP アクセスを実質的に無効にします。 |
| ステップ 4 | UCS-A /system/services # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

**例**

次に、HTTP から HTTPS へのリダイレクションを有効にし、トランザクションをコミットする例を示します。

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # enable http-redirect
Warning: When committed, this closes all the web sessions.
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

# ネットワーク関連のサービス

## SNMP 機能の概要

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： Cisco UCS内のソフトウェア コンポーネントです。 Cisco UCSのデータを維持し、必要に応じて SNMP マネージャにレポートします。Cisco UCS にはエージェントと MIB のコレクションが含まれます。SNMP エージェントをイネーブルにして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、Cisco UCS Manager で SNMP をイネーブルにして設定します。
- Managed Information Base（MIB）：SNMP エージェントの管理対象オブジェクトの集合。Cisco UCS リリース 1.4(1) 以降では、それ以前のリリースより大量の MIB をサポートしています。

Cisco UCS 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）

# 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。

Cisco UCS Manager 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、Cisco UCS Manager 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します。Cisco UCS Manager 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。

# 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 モデルは選択されたセキュリティ レベルと組み合わされ、SNMP メッセージの処理中に適用されるセキュリティ メカニズムを決定します。

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。権限のレベルによって、メッセージが情報開示の保護を必要とするか、またはメッセージが認証されるかが決定されます。サポートされるセキュリティ レベルは、実装されているセキュリティ モデル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 | レベル | 認証 | 暗号化 | 結果 |
|---|---|---|---|---|
| v1 | noAuthNoPriv | コミュニティ ストリング | 未対応 | 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 | 未対応 | 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 | 未対応 | ユーザ名の照合を使用して認証します。 |
| v3 | authNoPriv | HMAC-MD5 または HMAC-SHA | 非対応 | Hash-Based Message Authentication Code（HMAC）メッセージ ダイジェスト 5（MD5）アルゴリズムまたは HMAC Secure Hash Algorithm（SHA）アルゴリズムに基づいて認証します。 |
| v3 | authPriv | HMAC-MD5 または HMAC-SHA | DES | HMAC-MD5 アルゴリズムまたは 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、管理操作および暗号化 SNMP メッセージを実行するために、設定されているユーザのみを承認します。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、データ シーケンスが悪意なく起こり得る範囲を超えて変更されていないことを保証します。
- メッセージの発信元の認証：メッセージ送信者の ID を確認できることを保証します。
- 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。

# Cisco UCS での SNMP サポート

Cisco UCS は、SNMP に対して以下のサポートを提供します。

### MIB のサポート

Cisco UCS は、MIB への読み取り専用アクセスをサポートします。

Cisco UCS で使用可能な特定の MIB およびその入手先については、B シリーズ サーバは http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/mib/b-series/b_UCS_MIBRef.html を、C シリーズは http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/mib/c-series/b_UCS_Standalone_C-Series_MIBRef.html を参照してください。

### SNMPv3 ユーザの認証プロトコル

Cisco UCS は、SNMPv3 ユーザに次の認証プロトコルをサポートします。

- HMAC-MD5-96（MD5）
- HMAC-SHA-96（SHA）

Cisco UCS Manager SNMPv3 が連邦情報処理標準 (FIPS) モードの場合、リリース 3.2(3) 以降のリリースでは MD5 認証がサポートされません。したがって、MD5 認証を持つの既存のまたは新しく作成されたにある SNMPv3 ユーザがこれらのリリースで導入されるしないと、次のエラー メッセージが表示されます。

```
Major     F1036    2018-02-01T14:36:32.995     99095 SNMP User testuser can't be
deployed. Error: MD5 auth is not supported
```

このようなユーザを展開するには、認証タイプを **SHA**に変更します。

### SNMPv3 ユーザの AES プライバシー プロトコル

Cisco UCS は、SNMPv3 メッセージ暗号化用のプライバシー プロトコルの 1 つとして Advanced Encryption Standard（AES）を使用し、RFC 3826 に準拠しています。

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 ユーザ用のプライバシー パスワードを含めると、Cisco UCS Manager 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 プライバシー パスワードは最小で 8 文字です。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。

Cisco UCS Manager リリース 3.2(3) 以降のリリースでは、AES 暗号化なしの SNMPv3 ユーザはサポートされていません。したがって、AES 暗号化を使用していない既存または新規に作成された SNMPv3 ユーザは、これらのリリースでは展開されず、次のようなエラーメッセージが表示されます。

```
Major     F1036    2018-02-01T14:36:32.995     99095 SNMP User testuser can't be
deployed. Error: AES is not enabled
```

このようなユーザを展開するには、 **aes-128**暗号化を有効にします。

# SNMP のイネーブル化および SNMP プロパティの設定

Cisco UCS ドメインからの SNMP メッセージには、システム名ではなくファブリック インターコネクト名が表示されます。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope monitoring** | モニタリング モードを開始します。 |
| **ステップ 2** | UCS-A /monitoring # **enable snmp** | SNMP を有効にします。 |
| **ステップ 3** | UCS-A /monitoring # **set snmp community** | snmp コミュニティ モードを開始します。 |
| **ステップ 4** | UCS-A /monitoring # **Enter a snmp community:** *community-name* | SNMP コミュニティを指定します。パスワードとしてコミュニティ名を使用します。コミュニティ名は、最大 32 文字の英数字で指定できます。 |
| **ステップ 5** | UCS-A /monitoring # **set snmp syscontact** *system-contact-name* | 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。 |
| **ステップ 6** | UCS-A /monitoring # **set snmp syslocation** *system-location-name* | 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。 |
| **ステップ 7** | UCS-A /monitoring # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

### 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。

```
UCS-A# scope monitoring
UCS-A /monitoring # enable snmp
UCS-A /monitoring* # set snmp community
UCS-A /monitoring* # Enter a snmp community: SnmpCommSystem2
UCS-A /monitoring* # set snmp syscontact contactperson1
UCS-A /monitoring* # set snmp syslocation systemlocation
UCS-A /monitoring* # commit-buffer
UCS-A /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

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope monitoring** | モニタリング モードを開始します。 |
| **ステップ 2** | UCS-A /monitoring # **enable snmp** | SNMP を有効にします。 |
| **ステップ 3** | UCS-A /monitoring # **create snmp-trap** {*hostname* \| *ip-addr* \| *ip6-addr*} | 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 ホストを作成します。<br>ホスト名は IPv4 アドレスの完全修飾ドメイン名にすることができます。 |
| **ステップ 4** | UCS-A /monitoring/snmp-trap # **set community** *community-name* | SNMP トラップに使用する SNMP コミュニティ名を指定します。 |
| **ステップ 5** | UCS-A /monitoring/snmp-trap # **set port** *port-num* | SNMP トラップに使用するポートを指定します。 |
| **ステップ 6** | UCS-A /monitoring/snmp-trap # **set version** {**v1** \| **v2c** \| **v3**} | 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。 |
| **ステップ 7** | （任意） UCS-A /monitoring/snmp-trap # **set notification type** {**traps** \| **informs**} | 送信するトラップのタイプ。ここに表示される値は次のとおりです。<br>• バージョンで v2c または v3 を選択した場合は **traps**。<br>• バージョンに v2c を選択する場合は **informs**。<br>（注） バージョンとして v2c を選択した場合にのみインフォーム通知を送信できます。 |
| **ステップ 8** | （任意） UCS-A /monitoring/snmp-trap # **set v3 privilege** {**auth** \| **noauth** \| **priv**} | バージョンに v3 を選択した場合、トラップに関連付けられた権限。<br>ここに表示される値は次のとおりです。<br>• **auth**：認証あり、暗号化なし |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| | | • **noauth**：認証なし、暗号化なし<br>• **priv**：認証あり、暗号化あり |
| ステップ 9 | UCS-A /monitoring/snmp-trap # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

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。

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

次の例は、SNMP を有効にし、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コミットします。

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

# SNMP トラップの削除

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope monitoring** | モニタリング モードを開始します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 2 | UCS-A /monitoring # **delete snmp-trap** {*hostname* \| *ip-addr*} | 指定したホスト名または IP アドレスの指定した SNMP トラップ ホストを削除します。 |
| ステップ 3 | UCS-A /monitoring # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

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。

```
UCS-A# scope monitoring
UCS-A /monitoring # delete snmp-trap 192.168.100.112
UCS-A /monitoring* # commit-buffer
UCS-A /monitoring #
```

# SNMPv3 ユーザの作成

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope monitoring** | モニタリング モードを開始します。 |
| ステップ 2 | UCS-A /monitoring # **enable snmp** | SNMP を有効にします。 |
| ステップ 3 | UCS-A /monitoring # **create snmp-user** *user-name* | 指定された SNMPv3 ユーザを作成します。<br>SNMP ユーザ名は、ローカル ユーザ名と同じにはできません。ローカル ユーザ名と一致しない SNMP ユーザ名を選択します。 |
| ステップ 4 | UCS-A /monitoring/snmp-user # **set aes-128** {**no** \| **yes**} | AES-128 暗号化の使用を有効または無効にします。 |
| ステップ 5 | UCS-A /monitoring/snmp-user # **set auth** {**md5** \| **sha**} | MD5 または DHA 認証の使用を指定します。 |
| ステップ 6 | UCS-A /monitoring/snmp-user # **set password** | ユーザ パスワードを指定します。 **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。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 7 | UCS-A /monitoring/snmp-user # **set priv-password** | ユーザ プライバシー パスワードを指定します。 **set priv-password** コマンドを入力すると、プライバシー 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 8 | UCS-A /monitoring/snmp-user # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

### 例

次の例は、SNMP を有効にし、snmp-user14 という名前の SNMPv3 ユーザを作成し、AES-128 暗号化を無効にし、MD5 認証の使用を指定し、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。

```
UCS-A# scope monitoring
UCS-A /monitoring # enable snmp
UCS-A /monitoring* # create snmp-user snmp-user14
UCS-A /monitoring/snmp-user* # set aes-128 no
UCS-A /monitoring/snmp-user* # set auth md5
UCS-A /monitoring/snmp-user* # set password
Enter a password:
Confirm the password:
UCS-A /monitoring/snmp-user* # set priv-password
Enter a password:
Confirm the password:
UCS-A /monitoring/snmp-user* # commit-buffer
UCS-A /monitoring/snmp-user #
```

# SNMPv3 ユーザの削除

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope monitoring** | モニタリング モードを開始します。 |
| ステップ 2 | UCS-A /monitoring # **delete snmp-user** *user-name* | 指定した SNMPv3 ユーザを削除します。 |
| ステップ 3 | UCS-A /monitoring # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

### 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。

```
UCS-A# scope monitoring
UCS-A /monitoring # delete snmp-user snmp-user14
UCS-A /monitoring* # commit-buffer
UCS-A /monitoring #
```

# Telnet のイネーブル化

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope system** | システム モードを開始します。 |
| ステップ 2 | UCS-A /system # **scope services** | システム サービス モードを開始します。 |
| ステップ 3 | UCS-A /services # **enable telnet-server** | Telnet サービスを有効にします。 |
| ステップ 4 | UCS-A /services # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

例

次に、Telnet を有効にし、トランザクションをコミットする例を示します。

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /services # enable telnet-server
UCS-A /services* # commit-buffer
UCS-A /services #
```

# CIMC Web サービスのイネーブル化

CIMC Web サービスを有効にするには：

- admin 権限でログインする必要があります。
- CIMC Web サービスは、デフォルトでは有効なので、無効にする必要があります。

手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope system** / | システム モードを開始します。 |
| ステップ 2 | UCS-A /system #**scope services**/ | システムのサービス モードを開始します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 3 | UCS-A/system/services #**enable cimcwebsvc**/ | CIMC Web サービスを有効にします。 |
| ステップ 4 | UCS-A/system/services *# **commit-buffer**/ | トランザクションをシステムの設定にコミットします。 |

### 例

次に、CIMC Web サービスを有効にし、トランザクションを保存する例を示します。

```
UCS-A# scope system
UCS-A/system # scope services
UCS-A/system/services # enable cimcwebsvc
UCS-A/system/services *# commit-buffer
UCS-A/system/services # commit-buffer
UCS-A/system/services # show cimcwebsvc
Name: cimcwebservice
    Admin State: Enabled
```

## CIMC Web サービスの無効化

CIMC Web サービスを無効にするには：

- admin 権限でログインする必要があります。
- CIMC Web サービスを有効にする必要があります。

（注） CIMC Web サービスはデフォルトで有効となっています。

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | UCS-A# **scope system** / | システム モードを開始します。 |
| ステップ 2 | UCS-A /system #**scope services**/ | システムのサービス モードを開始します。 |
| ステップ 3 | UCS-A/system/services #**disable cimcwebsvc**/ | CIMC Web サービスを無効にします。 |
| ステップ 4 | UCS-A/system/services *# **commit-buffer**/ | トランザクションをシステムの設定にコミットします。 |

### 例

次に、CIMC Web サービスを無効にし、トランザクションを保存する例を示します。

```
UCS-A# scope system
UCS-A/system # scope services
UCS-A/system/services # disable cimcwebsvc
UCS-A/system/services *# commit-buffer
UCS-A/system/services # commit-buffer
UCS-A/system/services # show cimcwebsvc
Name: cimcwebservice
    Admin State: Disabled
```

# 通信サービスのディセーブル化

### 手順

| | コマンドまたはアクション | 目的 |
|---|---|---|
| **ステップ 1** | UCS-A# **scope system** | システム モードを開始します。 |
| **ステップ 2** | UCS-A /system # **scope services** | システム サービス モードを開始します。 |
| **ステップ 3** | UCS-A /system/services # **disable** *service-name* | 指定したサービスを無効にします。ここで *service-name* 引数は次のいずれかのキーワードです。<br>• *cimxml* ：CIM XML サービスを無効にします。<br>• **http** ：HTTP サービスを無効にします。<br>• **https** ：HTTPS サービスを無効にします。<br>• **telnet-server** ：Telnet サービスを無効にします。 |
| **ステップ 4** | UCS-A /system/services # **commit-buffer** | トランザクションをシステムの設定にコミットします。 |

### 例

次に、CIM XML 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。

```
UCS-A# scope system
UCS-A# scope services
UCS-A /system/services # disable cimxml
```

```
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```