管理アクセス

この章では、Telnet、SSH、およびHTTPS（ASDMを使用）経由でシステム管理のためにASAにアクセスする方法、ユーザーを認証および許可する方法、およびログインバナーを作成する方法について説明します。

管理リモート アクセスの設定

ここでは、ASDM 用の ASA アクセス、Telnet または SSH、およびログイン バナーなどのその他のパラメータの設定方法について説明します。

SSH アクセスの設定

SSH ガイドライン

- また、ASA インターフェイスに SSH アクセスの目的でアクセスするために、ホスト IP アドレスを許可するアクセス ルールは必要ありません。このセクションの手順に従って、SSH アクセスを設定する必要があるだけです。
- ASA への通過ルートとなるインターフェイス以外のインターフェイスへの SSH アクセスはサポートされません。たとえば、SSH ホストが外部インターフェイスにある場合、外部インターフェイスへの直接管理接続のみ開始できます。
- ASA は、コンテキスト/単一のモードあたり最大 5 つの同時 SSH 接続と、すべてのコンテキストにまたがり分散された最大 100 の接続を許容します。ただし、設定コマンドは変更されるリソースをロックする可能性があるため、すべての変更が正しく適用されるように、一度に 1 つの SSH セッションで変更を行う必要があります。
- SSH では次の機能はサポートされていません。

- EDDSA キーペア
- FIPS モードの RSA キーペア

- Cisco ASA **copy** コマンドを使用して SCP サーバーとの間でファイルをコピーするには、次の手順を実行する必要があります。
  - **ssh** コマンドを使用して、Cisco ASA で SCP サーバーサブネット/ホストの SSH アクセスを有効にします。
  - **crypto key generate**コマンドを使用してキーペアを生成します（物理 Cisco ASA の場合のみ）。
- SSH デフォルトユーザー名はサポートされなくなりました。**pix** または **asa** ユーザー名とログイン パスワードで SSH を使用して ASA に接続することができなくなりました。SSH を使用するには、**aaa authentication ssh console LOCAL** コマンドを使用して AAA 認証を設定してから、**username** コマンドを入力してローカル ユーザーを定義します。ローカル データベースの代わりにAAAサーバーを認証に使用する場合、ローカル認証もバックアップの手段として設定しておくことをお勧めします。
- SSH バージョン 2 のみがサポートされます。

Cisco ASA への SSH アクセスを構成するには、SSH サーバーをイネーブルにして、許可する IP アドレスを指定します。ユーザーを認証するために、次の方法を使用できます。

- ローカルデータベースまたは AAA サーバーのいずれかを使用するユーザー名とパスワード。
- ローカルデータベースを使用したユーザー名と公開キー。
- X.509v3 証明書（ユーザー名は証明書から派生）およびローカルデータベースまたはAAAサーバーからの承認。

デフォルトでは、SSH クライアントでサポートされているアルゴリズムに応じて、X.509 証明書または公開キー認証方式のいずれかが試行されますが、同じセッションで両方が試行されることはありません。X.509 認証も公開キー認証も成功しなかった場合、Cisco ASA はパスワード認証を試みます。必要に応じて、メソッドを禁止できます。

SSH サーバーの有効化

SSH サーバーを有効にし、接続を許可する IP アドレスを指定します。その他の SSH サーバー設定を行うこともできます。

始める前に

- マルチ コンテキスト モードでは、コンテキスト実行スペースで次の手順を実行します。システムからコンテキスト コンフィギュレーションに変更するには、**changeto context** *name* を入力します。

手順

ステップ 1 ASA がアドレスまたはサブネットごとに接続を受け入れる IP アドレスと、SSH を使用可能なインターフェイスを特定します。

ssh *source_IP_address mask source_interface*

- *source_interface*：名前付きインターフェイスを指定します。ブリッジ グループの場合、ブリッジ グループ メンバ インターフェイスを指定します。

Telnet と異なり、SSH は最も低いセキュリティ レベルのインターフェイスで実行できます。

例：

```
ciscoasa(config)# ssh 192.168.3.0 255.255.255.0 inside
```

ステップ 2 （任意） ASA がセッションを切断するまでに SSH がアイドル状態を維持する時間の長さを設定します。

ssh timeout *minutes*

例：

```
ciscoasa(config)# ssh timeout 30
```

タイムアウトは 1 ～ 60 分に設定します。デフォルトは 5 分です。デフォルトの期間では一般に短すぎるので、実働前のテストとトラブルシューティングがすべて完了するまでは、長めに設定しておいてください。

ステップ 3 （任意） 1 つ以上の認証方式を禁止します。

ssh authentication method {[**publickey**] [**x509-certificate**] [**password**]}

デフォルトでは、SSH クライアントでサポートされているアルゴリズムに応じて、X.509 証明書または公開キー認証方式のいずれかが試行されますが、同じセッションで両方が試行されることはありません。X.509 認証も公開キー認証も成功しなかった場合、Cisco ASA はパスワード認証を試みます。

使用しない方式を省略します。コマンド内の方式の順序は、試行される方式の順序には影響しません。

例：

パスワード認証を禁止するには、次のように設定します。

```
ciscoasa(config)# ssh authentication method publickey x509-certificate
```

ステップ 4 （任意） セキュアコピー（SCP）サーバーをイネーブルにできます。

ssh scopy enable

SCP サーバーにはディレクトリサポートがありません。ディレクトリ サポートがないため、ASA の内部ファイルへのリモート クライアント アクセスは制限されます。

SCP サーバーでは、バナーまたはワイルドカードがサポートされていません。

ステップ 5 （任意） SSH 暗号の暗号化アルゴリズムを設定します。

ssh cipher encryption {**all** | **fips** | **high** | **low** | **medium** | **custom** *colon-delimited_list_of_encryption_ciphers*}

例：

```
ciscoasa(config)# ssh cipher encryption custom 3des-cbc:aes128-cbc:aes192-cbc
```

デフォルトは**medium**です。暗号方式は、リストされた順に使用されます。事前定義されたリストでは、暗号方式が最も高いの順で、最も低いセキュリティに割り当てられています。

- **all** キーワードは、すべての暗号（AEAD_AES_256_GCM aes256-gcm@openssh.com 3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes128-gcm@openssh.com chacha20-poly1305@openssh.com aes192-ctr aes256-ctr aes256-gcm@openssh.com）を使用することを指定します
- カスタム暗号ストリングを設定する場合は、**custom** キーワードを使用し、各暗号ストリングをコロンで区切って入力します。
- **fips** キーワードは、FIPS に準拠する暗号（AEAD_AES_256_GCM aes256-gcm@openssh.com aes128-cbc aes256-cbc）のみを使用することを指定します
- **high** キーワードは、非常に強力な暗号（AEAD_AES_256_GCM aes256-gcm@openssh.com aes256-cbc aes128-gcm@openssh.com chacha20-poly1305@openssh.com aes256-ctr aes256-gcm@openssh.com）のみを使用することを指定します
- **low** キーワードは、低、中、高程度の暗号（3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr AEAD_AES_256_GCM aes256-gcm@openssh.com）を使用することを指定します
- **medium** キーワードはデフォルト値で、中、高程度の暗号（3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr AEAD_AES_256_GCM aes256-gcm@openssh.com）を使用することを指定します

ステップ 6 （任意） SSH 暗号の整合性アルゴリズムを設定します。

ssh cipher integrity {**all** | **fips** | **high** | **low** | **medium** | **custom** *colon-delimited_list_of_integrity_ciphers*}

例：

```
ciscoasa(config)# ssh cipher integrity custom hmac-sha1-96:hmac-md5
```

デフォルトは **high** です。

- すべての暗号方式（hmac-sha1 hmac-sha1-96（非推奨）hmac-md5（非推奨）hmac-md5-96（非推奨）hmac-sha2-256）を使用する場合は、**all** キーワードを使用します。

- カスタム暗号ストリングを設定する場合は、**custom** キーワードを使用し、各暗号ストリングをコロンで区切って入力します。
- FIPS 対応の暗号方式（hmac-sha1 hmac-sha2-256）のみを使用する場合は、**fips** キーワードを使用します。
- 強度が高の暗号方式のみ（hmac-sha2-256）を使用する場合は、**high** キーワードを使用します（デフォルト）。
- 強度が低、中、高の暗号方式（hmac-sha1 hmac-sha1-96 hmac-md5 hmac-md5-96 hmac-sha2-256）を使用する場合は、**low** キーワードを使用します。
- 強度が中および高の暗号方式（hmac-sha1 hmac-sha1-96（非推奨）hmac-sha2-256）を使用する場合は、**medium** キーワードを使用します。

ステップ 7 （任意）（管理コンテキストのみ）Diffie-Hellman（DH）キー交換モードを設定します。

ssh key-exchange group {**curve25519-sha256** | **dh-group14-sha1** | **dh-group14-sha256** | **ecdh-sha2-nistp256**}

例：

```
ciscoasa(config)# ssh key-exchange group dh-group14-sha1
```

デフォルトは **dh-group14-sha256**

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。キー交換は管理コンテキストでのみ設定できます。この値はすべてのコンテキストで使用されます。

例

次に、ASA への SCP セッションの例を示します。外部ホストのクライアントから、SCP ファイル転送を実行します。たとえば、Linux では次のコマンドを入力します。

scp -v -pw *password* [*path*/]*source_filename*
username@asa_address:{**disk0**|**disk1**}:/[*path*/]*dest_filename*

-v は冗長を表します。**-pw** が指定されていない場合は、パスワードの入力を求めるプロンプトが表示されます。

パスワードアクセス用の SSH の設定

ユーザー名とパスワードを使用して SSH 認証を設定します。

始める前に

- マルチ コンテキスト モードでは、コンテキスト実行スペースで次の手順を実行します。システムからコンテキスト コンフィギュレーションに変更するには、**changeto context** *name* を入力します。

手順

ステップ 1 SSH に必要なキーペアを生成します（物理 ASA の場合のみ）。

ASA 仮想 の場合、キーペアは導入後に自動的に作成されます。

a) デフォルトキーペアを生成します。

crypto key generate {**ecdsa elliptic-curve** *size* |**rsa modulus** *size*}

例：

```
ciscoasa(config)# crypto key generate ecdsa elliptic-curve 521
```

- **ecdsa elliptic-curve** *size*：ビット単位のサイズは 256、384、または 521 です。
- **rsa modulus** *size*：ビット単位のサイズは 2048、3072、または 4096 です。

SSH によって使用されるキーは <Default-*type*-Key> と呼ばれます。**label** キーワードを指定しないでください。デフォルト以外のキーペアは使用しません。指定するキーのサイズが大きいほど、キーペアの生成にかかる時間は長くなります。SSH は ECDSA、RSA の順にキーを試みます。**show crypto key mypubkey** {**ecdsa** | **rsa**} コマンドを使用してキーを表示します。

b) （任意） デフォルトのキー順序（ECDSA、RSA）を使用しない場合は、使用するキーペアを指定します。

ssh key-exchange hostkey {**rsa** | **ecdsa**}

RSA を選択した場合、2048 以上のキーサイズを使用する必要があります。アップグレードの互換性のために、これより小さいキーは、デフォルトのキー順序を使用する場合にのみサポートされます。

例：

```
ciscoasa(config)# ssh key-exchange hostkey ecdsa
```

ステップ 2 キーを永続的なフラッシュメモリに保存します。

write memory

例：

```
ciscoasa(config)# write memory
```

ステップ 3 SSH アクセスに使用できるローカルデータベースまたは AAA サーバーでユーザーを作成します。ローカルユーザー名を追加するには、次のコマンドを参照してください（推奨）。

username *name* **password** *password* **privilege** *level*

同じローカルユーザーに対して、公開キー認証またはパスワード認証を使用できます。AAA サーバーを使用した公開キー認証はサポートされていません。

例：

```
ciscoasa(config)# username admin password Far$cape1999 privilege 15
```

デフォルトの特権レベルは 2 です。0 ～ 15 の範囲でレベルを入力します。15 を指定すると、すべての特権を使用できます。

ステップ 4 SSH アクセスのローカルまたは AAA サーバー認証をイネーブルにします。

aaa authentication ssh console {**LOCAL** | *server_group* [**LOCAL**]}

このコマンドは、**ssh authentication** コマンドセットでのユーザー名のローカル公開キー認証には影響しません。ASA では、公開キー認証に対し、ローカル データベースを暗黙的に使用します。このコマンドは、ユーザー名とパスワードにのみ影響します。ローカルユーザーが公開キー認証またはパスワードを使用できるようにするには、パスワードアクセスを有効にするため、このコマンドで明示的にローカル認証を設定する必要があります。

例：

```
ciscoasa(config)# aaa authentication ssh console LOCAL
```

公開キーアクセス用の SSH の設定

公開キーを使用して SSH 認証を設定します。

始める前に

- マルチ コンテキスト モードでは、コンテキスト実行スペースで次の手順を実行します。システムからコンテキスト コンフィギュレーションに変更するには、**changeto context** *name* を入力します。

手順

ステップ 1 SSH に必要なキーペアを生成します（物理 ASA の場合のみ）。

ASA 仮想 の場合、キーペアは導入後に自動的に作成されます。

a) デフォルトキーペアを生成します。
crypto key generate {**ecdsa elliptic-curve** *size* |**rsa modulus** *size*}

例：

```
ciscoasa(config)# crypto key generate ecdsa elliptic-curve 521
```

- **ecdsa elliptic-curve** *size*：ビット単位のサイズは 256、384、または 521 です。
- **rsa modulus** *size*：ビット単位のサイズは 2048、3072、または 4096 です。

SSH によって使用されるキーは <Default-*type*-Key> と呼ばれます。**label** キーワードを指定しないでください。デフォルト以外のキーペアは使用しません。指定するキーのサイズが大きいほど、キーペアの生成にかかる時間は長くなります。SSH は ECDSA、RSA の順にキーを試みます。**show crypto key mypubkey** {**ecdsa** | **rsa**} コマンドを使用してキーを表示します。

b) （任意） デフォルトのキー順序（ECDSA、RSA）を使用しない場合は、使用するキーペアを指定します。

ssh key-exchange hostkey {**rsa** | **ecdsa**}

RSA を選択した場合、2048 以上のキーサイズを使用する必要があります。アップグレードの互換性のために、これより小さいキーは、デフォルトのキー順序を使用する場合にのみサポートされます。

例：

```
ciscoasa(config)# ssh key-exchange hostkey ecdsa
```

ステップ 2 キーを永続的なフラッシュメモリに保存します。

write memory

例：

```
ciscoasa(config)# write memory
```

ステップ 3 SSH アクセスに使用できるユーザーをローカル データベースに作成します。

username *name* [**password** *password*] **privilege** *level*

例：

```
ciscoasa(config)# username admin password Far$cape1999 privilege 15
```

デフォルトの特権レベルは 2 です。0 ～ 15 の範囲でレベルを入力します。15 を指定すると、すべての特権を使用できます。ユーザーにパスワード認証ではなく公開キー認証（**ssh authentication**）を強制する場合は、パスワードなしでユーザーを作成することを推奨します。**username** コマンドで公開キー認証およびパスワードの両方を設定した場合、ユーザーはいずれの方法でもログインできます（パスワードアクセス用の SSH の設定 （5 ページ）で AAA 認証を明示的に設定した場合）。**注**：ユーザー名とパスワードを作成しなければならないとい

う事態を回避するため、**username** コマンド **nopassword** オプション**nopassword** オプションでは、任意のパスワードを入力できますが、パスワードなしは不可能です。

ステップ 4 パスワード認証ではなく公開キー認証のみ、またはこれら両方の認証をユーザーに許可し、ASA で公開キーを入力します。

username *name* **attributes**

ssh authentication {**pkf** | **publickey** *key*}

例：

```
ciscoasa(config)# username admin attributes
ciscoasa(config-username)# ssh authentication pkf

Enter an SSH public key formatted file.
End with the word "quit" on a line by itself:
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "256-bit ED25519, converted by dean@dwinchester-mac from "
AAAAC3NzaC1lZDI1NTE5AAAAIDmIeTNfEOnuH0094p1MKX80fW2O216g4trnf7gwWe5Q
---- END SSH2 PUBLIC KEY ----
quit
INFO: Import of an SSH public key formatted file SUCCEEDED.
```

ローカル **username** の場合、パスワード認証ではなく公開キー認証のみ、またはこれら両方の認証を有効にできます。ssh-rsa、または ecdsa-sha2-nistp raw キー（証明書なし）を生成可能な任意の SSH キー生成ソフトウェア（ssh keygen など）を使用して、公開キー/秘密キーのペアを生成できます。ASA で公開キーを入力します。その後、SSH クライアントは秘密キー（およびキー ペアを作成するために使用したパスフレーズ）を使用して ASA に接続します。

pkf キーの場合、PKF でフォーマットされたキーを最大 4096 ビット貼り付けるよう求められます。Base64 形式では大きすぎてインラインで貼り付けることができないキーにはこのフォーマットを使用します。たとえば、ssh keygen を使って 4096 ビットのキーを生成してから PKF に変換し、そのキーに対して **pkf** キーワードが求められるようにすることができます。**注**：フェールオーバーで **pkf** オプションを使用することはできますが、PKF キーは、スタンバイシステムに自動的に複製されません。PKF キーを同期するには、**write standby** コマンドを入力する必要があります。

publickey キーの場合、これは Base64 でエンコードされた公開キーのことです。ssh-rsa、または ecdsa-sha2-nistp raw キー（証明書なし）を生成可能な任意の SSH キー生成ソフトウェア（ssh keygen など）を使用して、キーを生成できます。

例

次に、PKF 形式のキーを使用して認証する例を示します。

```
ciscoasa(config)# crypto key generate rsa modulus 4096
ciscoasa(config)# write memory
ciscoasa(config)# username exampleuser1 password examplepassword1 privilege 15
ciscoasa(config)# username exampleuser1 attributes
ciscoasa(config-username)# ssh authentication pkf
```

```
Enter an SSH public key formatted file.
End with the word "quit" on a line by itself:
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by xxx@xxx from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAACAQDNUvkgza37lB/Q/fljpLAv1BbyAd5PJCJXh/U4LO
hleR/qgIROjpnFaS7Az8/+sjHmq0qXC5TXkzWihvRZbhefyPhPHCi0hIt4oUF2ZbXESA/8
jUT4ehXIUE7FrChffBBtbD4d9FkV8A2gwZCDJBxEM26ocbZCSTx9QC//wt6E/zRcdoqiJG
p4ECEdDaM+56l+yf73NUigO7wYkqcrzjmI1rZRDLVcqtj8Q9qD3MqsV+PkJGSGiqZwnyIl
QbfYxXHU9wLdWxhUbA/xOjJuZ15TQMa7KLs2u+RtrpQgeTGTffIh6O+xKh93gwTgzaZTK4
CQ1kuMrRdNRzza0byLeYPtSlv6Lv6F6dGtwlqrX5a+w/tV/aw9WUg/rapekKloz3tsPTDe
p866AFzU+Z7pVR1389iNuNJHQS7IUA2m0cciIuCM2we/tVqMPYJl+xgKAkuHDkBlMS4i8b
Wzyd+4EUMDGGZVeO+corKTLWFO1wIUieRkrUaCzjComGYZdzrQT2mXBcSKQNWlSCBpCHsk
/r5uTGnKpCNWfL7vd/sRCHyHKsxjsXR15C/5zgHmCTAaGOuIq0Rjo34+61+70PCtYXebxM
Wwm19e3eH2PudZd+rj1dedfr2/IrislEBRJWGLoR/N+xsvwVVM1Qqw1uL4r99CbZF9NghY
NRxCQOY/7K77II==
---- END SSH2 PUBLIC KEY ----
quit
INFO: Import of an SSH public key formatted file SUCCEEDED.
ciscoasa(config)#
```

次の例では、Linux または Macintosh システムの SSH の共有キーを生成して、ASA にインポートします。

1. コンピューターで 4,096 ビットの RSA 公開キーおよび秘密キーを生成します。

```
jcrichton-mac:~ john$ ssh-keygen -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/john/.ssh/id_rsa):
/Users/john/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase): pa$$phrase
Enter same passphrase again: pa$$phrase
Your identification has been saved in /Users/john/.ssh/id_rsa.
Your public key has been saved in /Users/john/.ssh/id_rsa.pub.
The key fingerprint is:
c0:0a:a2:3c:99:fc:00:62:f1:ee:fa:f8:ef:70:c1:f9 john@jcrichton-mac
The key's randomart image is:
+--[ RSA 4096]----+
| .               |
|  o  .           |
|+...  o          |
|B.+.....         |
|.B ..+  S        |
|  =   o          |
|   + . E         |
|  o o            |
| ooooo           |
+-----------------+
```

2. PKF 形式にキーを変換します。

```
jcrichton-mac:~ john$ cd .ssh
jcrichton-mac:.ssh john$ ssh-keygen -e -f id_rsa.pub
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by john@jcrichton-mac from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAACAQDNUvkgza37lB/Q/fljpLAv1BbyAd5PJCJXh/U4LO
hleR/qgIROjpnDaS7Az8/+sjHmq0qXC5TXkzWihvRZbhefyPhPHCi0hIt4oUF2ZbXESA/8
jUT4ehXIUE7FrChffBBtbD4d9FkV8A2gwZCDJBxEM26ocbZCSTx9QC//wt6E/zRcdoqiJG
p4ECEdDaM+56l+yf73NUigO7wYkqcrzjmI1rZRDLVcqtj8Q9qD3MqsV+PkJGSGiqZwnyIl
QbfYxXHU9wLdWxhUbA/xOjJuZ15TQMa7KLs2u+RtrpQgeTGTffIh6O+xKh93gwTgzaZTK4
```

```
CQ1kuMrRdNRzza0byLeYPtSlv6Lv6F6dGtwlqrX5a+w/tV/aw9WUg/rapekKloz3tsPTDe
p866AFzU+Z7pVR1389iNuNJHQS7IUA2m0cciIuCM2we/tVqMPYJl+xgKAkuHDkBlMS4i8b
Wzyd+4EUMDGGZVeO+corKTLWFO1wIUieRkrUaCzjComGYZdzrQT2mXBcSKQNWlSCBpCHsk
/r5uTGnKpCNWfL7vd/sRCHyHKsxjsXR15C/5zgHmCTAaGOuIq0Rjo34+61+70PCtYXebxM
Wwm19e3eH2PudZd+rj1dedfr2/IrislEBRJWGLoR/N+xsvwVVM1Qqw1uL4r99CbZF9NghY
NRxCQOY/7K77IQ==
---- END SSH2 PUBLIC KEY ----
jcrichton-mac:.ssh john$
```

3. キーをクリップボードにコピーします。
4. ASDM で、**[構成（Configuration）]** > **[デバイス管理（Device Management）]** > **[ユーザ/AAA（Users/AAA）]** > **[ユーザアカウント（User Accounts）]** の順に選択し、ユーザ名を選択してから [編集（Edit）] をクリックします。[Public Key Using PKF] をクリックして、ウィンドウにキーを貼り付けます。
5. ユーザが ASA に SSH できることを確認します。パスワードには、キーペアの作成時に指定した SSH キーパスワードを入力します。

```
jcrichton-mac:.ssh john$ ssh test@10.86.118.5
The authenticity of host '10.86.118.5 (10.86.118.5)' can't be established.
RSA key fingerprint is 39:ca:ed:a8:75:5b:cc:8e:e2:1d:96:2b:93:b5:69:94.
Are you sure you want to continue connecting (yes/no)? yes
```

次のダイアログボックスが、パスフレーズを入力するために表示されます。

一方、端末セッションでは、以下が表示されます。

```
Warning: Permanently added '10.86.118.5' (RSA) to the list of known hosts.
Identity added: /Users/john/.ssh/id_rsa (/Users/john/.ssh/id_rsa)
Type help or '?' for a list of available commands.
asa>
```

X.509 証明書アクセス用の SSH の設定

X.509v3 証明書による認証および承認を有効にします。

始める前に

- マルチ コンテキスト モードでは、コンテキスト実行スペースで次の手順を実行します。システムからコンテキスト コンフィギュレーションに変更するには、**changeto context** *name* を入力します。

手順

ステップ 1 トラストポイントを設定します。デジタル証明書の設定を参照してください。

ステップ 2 SSH クライアントを検証するトラストポイントを有効にします。

crypto ca trustpoint *trustpoint_name*

validation-usage ssh-client[**ipsec-client**] [**ssl-client**] [**ssl-server**]

さらにこれらのタイプを指定する場合は、このトラストポイントまたは IPsec クライアント、SSL クライアント、および SSL サーバーを使用することもできます。

例：

```
ciscoasa(config)# crypto ca trustpoint lets-encrypt_CA
ciscoasa(config-ca-trustpoint)# validation-usage ssh-client ipsec-client ssl-client
ssl-server
```

ステップ 3 証明書から取得されたユーザー名と一致するユーザーを認可サーバーに追加します。

- AAA サーバー：AAA サーバーおよびサーバーグループを参照して、AAA サーバーグループを Cisco ASA に追加します。
- ローカルデータベース：ローカル データベースへのユーザー アカウントの追加を参照してください。

このユーザーに対してもパスワード認証を許可する場合、ユーザーパスワードを設定できます。パスワード認証を設定しない場合、パスワードなしでユーザーを追加できます。この場合、パスワードを設定しても無視されます。

コンソール特権は、認可サーバーから返された属性に基づいて割り当てられます。ユーザーに設定する属性については、管理許可による CLI および ASDM アクセスの制限 （32 ページ）を参照してください。

例：

```
ciscoasa(config)# username dwinchester attributes
ciscoasa(config-username)# service-type admin
```

ステップ 4 SSH に使用するトラストポイントを識別します。

ssh trustpoint sign *trustpoint_name*

例：

```
ciscoasa(config)# ssh trustpoint sign ?
```

```
configure mode commands/options:
Available configured trustpoints:
  _SmartCallHome_ServerCA
  lets-encrypt_CA

ciscoasa(config)# ssh trustpoint sign lets-encrypt_CA
```

ステップ 5 Cisco ASA が証明書からユーザー名を取得する方法を定義します。

ssh username-from-certificate { [**cn**] [**upn**]}

1 つまたは両方のオプションを指定します。試行される順序は、コマンドのオプションの順序によって決まります。

- **cn**：共通名を使用します。
- **upn**：ユーザープリンシパル名を使用します。

例：

```
ciscoasa(config)# ssh username-from-certificate cn
```

ステップ 6 ローカルデータベースまたは AAA サーバーを使用してユーザーを認可します。

aaa authorization exec ssh-x509 {*server_group* | **LOCAL**} [**auto-enable**]

- **auto-enable**：十分な認証特権を持つ管理者が、ログインするときに特権 EXEC モードに自動的に入ることができます。

```
ciscoasa(config)# aaa authorization exec ssh-x509 LOCAL auto-enable
```

例

次の例では、ローカルデータベースを使用した認可で、SSH X.509 認証を有効にしています。

```
ciscoasa(config)# crypto ca trustpoint lets-encrypt_CA
ciscoasa(config-ca-trustpoint)# validation-usage ssh-client ipsec-client ssl-client
ssl-server
ciscoasa(config-ca-trustpoint)# exit
ciscoasa(config)# username dwinchester attributes
ciscoasa(config-username)# service-type admin
ciscoasa(config-username)# exit
ciscoasa(config)# aaa authorization exec ssh-x509 LOCAL auto-enable
ciscoasa(config)# ssh trustpoint sign lets-encrypt_CA
ciscoasa(config)# ssh username-from-certificate cn upn
```

Telnet アクセスの設定

Telnet を使用して ASA にアクセス可能なクライアント IP アドレスを指定するには、次の手順を実行します。次のガイドラインを参照してください。

- また、ASA インターフェイスに Telnet アクセスの目的でアクセスするために、ホスト IP アドレスを許可するアクセス ルールは必要ありません。このセクションの手順に従って、Telnet アクセスを設定する必要があるだけです。
- ASA への通過ルートとなるインターフェイス以外のインターフェイスへの Telnet アクセスはサポートされません。たとえば、Telnet ホストが外部インターフェイスにある場合、外部インターフェイスへの直接 Telnet 接続のみ開始できます。このルールの例外は、VPN 接続を介した場合のみです。VPN トンネルを介した管理アクセスの設定 （19 ページ）を参照してください。
- VPN トンネル内で Telnet を使用する場合を除き、最も低いセキュリティ インターフェイスに対して Telnet は使用できません。
- ASA は、コンテキスト/単一のモードあたり最大 5 つの同時 Telnet 接続と、すべてのコンテキストにまたがり分散された最大 100 の接続を許容します。

始める前に

- マルチ コンテキスト モードでは、コンテキスト実行スペースで次の手順を実行します。システムからコンテキスト コンフィギュレーションに変更するには、**changeto context** *name* を入力します。
- Telnet を使用して ASA CLI にアクセスするには、**password** コマンドで設定したログイン パスワードを入力します。Telnet を使用する前に手動でパスワードを設定する必要があります。

手順

ステップ 1 ASA が指定したインターフェイスのアドレスまたはサブネットごとに接続を受け入れる IP アドレスを特定します。

telnet *source_IP_address mask source_interface*

- *source_interface*：名前付きインターフェイスを指定します。ブリッジ グループの場合、ブリッジ グループ メンバインターフェイスを指定します。VPN 管理アクセスのみ（VPN トンネルを介した管理アクセスの設定 （19 ページ）を参照してください）の場合、名前付き BVI インターフェイスを指定します。

インターフェイスが 1 つしかない場合は、インターフェイスのセキュリティ レベルが 100 である限り、そのインターフェイスにアクセスするように Telnet を設定することができます。

例：

```
ciscoasa(config)# telnet 192.168.1.2 255.255.255.255 inside
```

ステップ 2 ASA がセッションを切断するまで Telnet セッションがアイドル状態を維持する時間の長さを設定します。

telnet timeout *minutes*

例：

```
ciscoasa(config)#  telnet timeout 30
```

タイムアウトは 1 ～ 1440 分に設定します。デフォルトは 5 分です。デフォルトの期間では一般に短すぎるので、実働前のテストとトラブルシューティングがすべて完了するまでは、長めに設定しておいてください。

例

次の例は、アドレスが 192.168.1.2 の内部インターフェイスのホストで ASA にアクセスする方法を示しています。

```
ciscoasa(config)# telnet 192.168.1.2 255.255.255.255 inside
```

次の例は、192.168.3.0 のネットワーク上のすべてのユーザーが内部インターフェイス上の ASA にアクセスできるようにする方法を示しています。

```
ciscoasa(config)# telnet 192.168.3.0. 255.255.255.255 inside
```

ASDM、その他のクライアントの HTTPS アクセスの設定

ASDM または CSM などの他の HTTPS クライアントを使用するには、HTTPS サーバーを有効にし、ASA への HTTPS 接続を許可する必要があります。HTTPS アクセスは工場出荷時のデフォルト設定の一部として有効化されています。HTTPSアクセスを設定するには、次のステップを実行します。次のガイドラインを参照してください。

- また、ASA インターフェイスに HTTPS アクセスの目的でアクセスするために、ホスト IP アドレスを許可するアクセス ルールは必要ありません。このセクションの手順に従って、HTTPSアクセスを設定する必要があるだけです。ただし、HTTP リダイレクトを設定して HTTP 接続を HTTPS に自動的にリダイレクトするには、HTTP を許可するアクセス ルールを有効化する必要があります。そうしないと、インターフェイスが HTTP ポートをリッスンできません。
- ASA への通過ルートとなるインターフェイス以外のインターフェイスへの管理アクセスはサポートされません。たとえば、管理ホストが外部インターフェイスにある場合、外部インターフェイスへの直接管理接続のみ開始できます。このルールの例外は、VPN 接続を介した場合のみです。VPN トンネルを介した管理アクセスの設定 （19 ページ）を参照してください。

- シングルコンテキストモードでは、最大 5 の ASDM 同時セッションを設定できます。マルチコンテキストモードでは、コンテキストごとに最大 5 つの同時 ASDM セッションを使用でき、全コンテキスト間で最大 200 の ASDM インスタンスの使用が可能です。

  ASDM セッションでは、2 つの HTTPS 接続が使用されます。一方は常に存在するモニター用で、もう一方は変更を行ったときにだけ存在する設定変更用です。たとえば、マルチコンテキスト モード システムの ASDM セッションの制限が 200 の場合、HTTPS セッション数は 400 に制限されます。

- ASA では、シングルコンテキストモードまたはコンテキストごとに最大 6 つの非 ASDM HTTPS 同時セッション（使用可能な場合）を許可し、すべてのコンテキスト間で最大または 100 の HTTPS セッションを許可します。

- 同じインターフェイス上で SSL（[webvpn] > [インターフェイスの有効化（enable interface）]）と HTTPS アクセスの両方を有効にした場合、**https://***ip_address* から セキュア クライアント にアクセスでき、**https://***ip_address***/admin** から ASDM にアクセスできます。どちらもポート 443 を使用します。**aaa authentication http console** も有効にする場合は、ASDM アクセス用に別のポートを指定する必要があります。

始める前に

- マルチ コンテキスト モードでは、コンテキスト実行スペースで次の手順を実行します。システムからコンテキスト コンフィギュレーションに変更するには、**changeto context** *name* を入力します。

手順

ステップ 1 ASA が指定したインターフェイスのアドレスまたはサブネットごとに HTTPS 接続を受け入れる IP アドレスを特定します。

http *source_IP_address mask source_interface*

- *source_interface*：名前付きインターフェイスを指定します。ブリッジグループの場合、ブリッジ グループ メンバインターフェイスを指定します。VPN 管理アクセスのみ（VPN トンネルを介した管理アクセスの設定 （19 ページ）を参照してください）の場合、名前付き BVI インターフェイスを指定します。

例：

```
ciscoasa(config)# http 192.168.1.2 255.255.255.255 inside
```

ステップ 2 HTTPS サーバーをイネーブルにします。

http server enable [*port*]

例：

```
ciscoasa(config)# http server enable 444
```

デフォルトでは、port は 443 です。ポート番号を変更する場合は、必ず ASDM アクセス URL に変更したポート番号を含めてください。たとえば、ポート番号を 444 に変更する場合は、次の URL を入力します。

https://10.1.1.1:444

ステップ 3 非ブラウザベースの HTTPS クライアントが ASA 上の HTTPS サービスにアクセスできるようにすることができます。デフォルトでは、ASDM、CSM、および REST API が許可されています。

http server basic-auth-client *user_agent*

- *user_agent*：HTTP 要求の HTTP ヘッダーにあるクライアントの User-Agent 文字列を指定します。完全な文字列または部分文字列を指定できます。部分文字列については、User-Agent 文字列の先頭と一致している必要があります。セキュリティを強化するために完全な文字列をお勧めします。文字列では大文字と小文字が区別されることに注意してください。

  たとえば、 **curl** は次の User-Agent 文字列と一致します。

  curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2

  curl は、次の User-Agent 文字列とは一致しません。

  abcd curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2

  CURL は、次の User-Agent 文字列とは一致しません。

  curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2

個別のコマンドを使用して、各クライアント文字列を入力します。多くの専門クライアント（python ライブラリ、curl、wget など）は、クロスサイト要求の偽造（CSRF）トークンベースの認証をサポートしていないため、これらのクライアントが ASA 基本認証方式を使用することを明確に許可する必要があります。セキュリティ上の理由から、必要なクライアントのみを許可する必要があります。

例：

```
ciscoasa(config)# http server basic-auth-client curl
```

ステップ 4 （任意） 接続とセッションのタイムアウトを設定します。

http server idle-timeout*minutes*

http server session-timeout*minutes*

http connection idle-timeout*seconds*

- **http server idle-timeout** *minutes*：ASDM 接続のアイドルタイムアウトを 1 ～ 1440 分の範囲で設定します。デフォルトは 20 分です。ASA は、設定した期間アイドル状態の ASDM 接続を切断します。
- **http server session-timeout** *minutes*：ASDM セッションのセッションタイムアウトを 1 ～ 1440 分の範囲で設定します。このタイムアウトはデフォルトで無効になっています。ASA は、設定した期間を超えた ASDM 接続を切断します。
- **http connection idle-timeout** *seconds*：ASDM、WebVPN、および他のクライアントを含むすべての HTTPS 接続のアイドルタイムアウトを 10 ～ 86400 秒の範囲で設定します。このタイムアウトはデフォルトで無効になっています。ASA は、設定した期間アイドル状態の接続を切断します。**http server idle-timeout** コマンドと **http connection idle-timeout** コマンドの両方を設定した場合は、**http connection idle-timeout** コマンドが優先されます。

例：

```
ciscoasa(config)# http server idle-timeout 30
ciscoasa(config)# http server session-timeout 120
```

例

次の例は、HTTPS サーバーを有効化し、アドレスが 192.168.1.2 の内部インターフェイス上のホストで ASDM にアクセスする方法を示しています。

```
ciscoasa(config)# http server enable
ciscoasa(config)# http 192.168.1.2 255.255.255.255 inside
```

次の例は、192.168.3.0/24 のネットワーク上のすべてのユーザーが内部インターフェイス上の ASDM にアクセスできるようにする方法を示しています。

```
ciscoasa(config)# http 192.168.3.0 255.255.255.0 inside
```

ASDM アクセスまたはクライアントレス SSL VPN のための HTTP リダイレクトの設定

ASDM またはクライアントレス SSL VPN を使用して ASA に接続するには、HTTPS を使用する必要があります。利便性のために、HTTP 管理接続を HTTPS にリダイレクトすることができます。たとえば、HTTP をリダイレクトすることによって、**http://10.1.8.4/admin/** または **https://10.1.8.4/admin/** と入力し、ASDM 起動ページで HTTPS アドレスにアクセスできます。

IPv4 と IPv6 の両方のトラフィックをリダイレクトできます。

始める前に

通常、ホスト IP アドレスを許可するアクセス ルールは必要ありません。ただし、HTTP リダイレクトのためには、HTTP を許可するアクセス ルールを有効化する必要があります。そうしないと、インターフェイスが HTTP ポートをリッスンできません。

手順

Enable HTTP redirect:

http redirect *interface_name* [*port*]

例：

```
ciscoasa(config)# http redirect outside 88
```

port は、インターフェイスが HTTP 接続のリダイレクトに使用するポートを指定します。デフォルトは 80 です。

VPN トンネルを介した管理アクセスの設定

あるインターフェイスで VPN トンネルが終端している場合、別のインターフェイスにアクセスして ASA を管理するには、そのインターフェイスを管理アクセス インターフェイスとして指定する必要があります。たとえば、外部インターフェイスから Cisco ASA に入る場合は、この機能を使用して、ASDMまたは Telnet 経由で内部インターフェイスに接続するか、外部インターフェイスから入るときに内部インターフェイスに ping を実行できます。

（注） この機能は SSH ではサポートされません。

（注） この機能は SNMP ではサポートされません。VPN 経由の SNMP の場合、ループバック インターフェイスで SNMP を有効にすることをお勧めします。ループバック インターフェイスで SNMP を使用するために、管理アクセス機能を有効にする必要はありません。ループバックは SSH でも機能します。

ASA への通過ルートとなるインターフェイス以外のインターフェイスへの VPN アクセスはサポートされません。たとえば、VPN アクセスが外部インターフェイスにある場合、外部インターフェイスへの直接接続のみ開始できます。複数のアドレスを覚える必要がないように、ASA の直接アクセス可能インターフェイスの VPN を有効にし、名前解決を使用してください。

管理アクセスは、IPsec クライアント、IPsec サイト間、Easy VPN、セキュアクライアント SSL VPN の VPN トンネルタイプ経由で行えます。

始める前に

- この機能は管理専用インターフェイスではサポートされません。
- 管理アクセスインターフェイスを使用し、アイデンティティ NAT を構成する場合、ルート ルックアップ オプションを使用して NAT を構成する必要があります。詳細については、適切なリリースの『ASA Firewall CLI Configuration Guide』の適切なリリースの「*NAT Examples and Reference*」の章の「NAT and VPN Management Access」の項を参照してください。

手順

別のインターフェイスから ASA に入るときにアクセスする管理インターフェイスの名前を指定します。

management-access *management_interface*

サイト間トンネルでは、名前付き BVI を指定できます（ルーテッドモード）。

例：

```
ciscoasa(config)# management-access inside
```

コンソール タイムアウトの変更

コンソール タイムアウトでは、接続を特権 EXEC モードまたはコンフィギュレーション モードにしておくことができる時間を設定します。タイムアウトに達すると、セッションはユーザー EXEC モードになります。デフォルトでは、セッションはタイムアウトしません。この設定は、コンソール ポートへの接続を保持できる時間には影響しません。接続がタイムアウトすることはありません。

手順

特権セッションが終了するまでのアイドル時間を分単位（0 ～ 60）で指定します。

console timeout *number*

例：

```
ciscoasa(config)# console timeout 0
```

デフォルトのタイムアウトは 0 であり、セッションがタイムアウトしないことを示します。

CLI プロンプトのカスタマイズ

プロンプトに情報を追加する機能により、複数のモジュールが存在する場合にログインしている ASA を一目で確認することができます。この機能は、フェールオーバー時に、両方の ASA に同じホスト名が設定されている場合に便利です。

マルチ コンテキスト モードでは、システム実行スペースまたは管理コンテキストにログインするときに、拡張プロンプトを表示できます。非管理コンテキスト内では、デフォルトのプロンプト（ホスト名およびコンテキスト名）のみが表示されます。

デフォルトでは、プロンプトに ASA のホスト名が表示されます。マルチ コンテキスト モードでは、プロンプトにコンテキスト名も表示されます。CLI プロンプトには、次の項目を表示できます。

cluster-unit	クラスタ ユニット名を表示します。クラスタの各ユニットは一意の名前を持つことができます。
コンテキスト	（マルチ モードのみ）現在のコンテキストの名前を表示します。
domain	ドメイン名を表示します。
hostname	ホスト名を表示します。
priority	フェールオーバー プライオリティを [pri]（プライマリ）または [sec]（セカンダリ）として表示します。
state	ユニットのトラフィック通過状態またはロールを表示します。 フェールオーバーの場合、state キーワードに対して次の値が表示されます。 • [act]：フェールオーバーが有効であり、装置ではトラフィックをアクティブに通過させています。 • [stby]：フェールオーバーはイネーブルです。ユニットはトラフィックを通過させていません。スタンバイ、失敗、または他の非アクティブ状態です。 • [actNoFailover]：フェールオーバーは無効であり、装置ではトラフィックをアクティブに通過させています。 • [stbyNoFailover]：フェールオーバーは無効であり、装置ではトラフィックを通過させていません。これは、スタンバイ ユニットでしきい値を上回るインターフェイス障害が発生したときに生じることがあります。 クラスタリングの場合、制御とデータの値が表示されます。

手順

次のコマンドを入力して、CLI プロンプトをカスタマイズします。

prompt {[**hostname**] [**context**] [**domain**] [**slot**] [**state**] [**priority**] [**cluster-unit**]}

例：

```
ciscoasa(config)# prompt hostname context slot state priority
ciscoasa/admin/pri/act(config)#
```

キーワードを入力する順序によって、プロンプト内の要素の順序が決まります。要素はスラッシュ（/）で区切ります。

ログイン バナーの設定

ユーザーが ASA に接続するとき、ログインする前、または特権 EXEC モードに入る前に表示されるメッセージを設定できます。

始める前に

- セキュリティの観点から、バナーで不正アクセスを防止することが重要です。「ウェルカム」や「お願いします」などの表現は侵入者を招き入れているような印象を与えるので使用しないでください。以下のバナーでは、不正アクセスに対して正しい表現を設定しています。

  ```
  You have logged in to a secure device.
  If you are not authorized to access this device,
  log out immediately or risk possible criminal consequences.
  ```

- バナーが追加された後、次の場合に ASA に対する Telnet または SSH セッションが終了する可能性があります。
  - バナー メッセージを処理するためのシステム メモリが不足している場合。
  - バナー メッセージの表示を試みたときに、TCP 書き込みエラーが発生した場合。
- バナー メッセージのガイドラインについては、RFC 2196 を参照してください。

手順

ユーザーが最初に接続したとき（「今日のお知らせ」（motd））、ユーザーがログインしたとき（login）、ユーザーが特権EXECモードにアクセスしたとき（exec）のいずれかに表示するバナーを追加します。

banner {**exec** | **login** | **motd**} *text*

例：

```
ciscoasa(config)# banner motd Only authorized access is allowed to $(hostname).
```

ユーザーが ASA に接続すると、まず「今日のお知らせ」バナーが表示され、その後にログイン バナーとプロンプトが表示されます。ユーザーが ASA に正常にログインすると、exec バナーが表示されます。

複数の行を追加する場合は、各行の前に **banner** コマンドを追加します。

バナー テキストに関する注意事項：

- スペースは使用できますが、CLI を使用してタブを入力することはできません。
- バナーの長さの制限は、RAM およびフラッシュ メモリに関するもの以外はありません。
- ASA のホスト名またはドメイン名は、**$(hostname)** 文字列と **$(domain)** 文字列を組み込むことによって動的に追加できます。
- システム コンフィギュレーションでバナーを設定する場合は、コンテキスト コンフィギュレーションで **$(system)** 文字列を使用することによって、コンテキスト内でそのバナー テキストを使用できます。

例

以下に、「今日のお知らせ」バナーを追加する例を示します。

```
ciscoasa(config)# banner motd Only authorized access is allowed to $(hostname).

ciscoasa(config)# banner motd Contact me at admin@example.com for any issues.
```

管理セッション クォータの設定

ASA で許可する ASDM、SSH、および Telnet の同時最大セッション数を設定できます。この最大値に達すると、それ以降のセッションは許可されず、syslog メッセージが生成されます。システム ロックアウトを回避するために、管理セッション割り当て量のメカニズムではコンソール セッションをブロックできません。

（注） マルチコンテキストモードではASDMセッションの数を設定することはできず、最大セッション数は 5 で固定されています。

（注） また、最大管理セッション（SSHなど）のコンテキストあたりのリソース制限を設定した場合は、小さい方の値が使用されます。

始める前に

マルチ コンテキスト モードでは、コンテキスト実行スペースで次の手順を実行します。システム コンフィギュレーションからコンテキスト コンフィギュレーションに切り替えるには、**changeto context** *name* コマンドを入力します。

手順

ステップ 1 次のコマンドを入力します。

quota management-session [**ssh** | **telnet** | **http** | **user**] *number*

- **ssh**：1 ～ 5 の SSH セッションの最大数を設定します。デフォルトは 5 分です。
- **telnet**：1 ～ 5 の Telnet セッションの最大数を設定します。デフォルトは 5 分です。
- **http**：1 ～ 5 の HTTPS（ASDM）セッションの最大数を設定します。デフォルトは 5 分です。
- **user**：1～5のユーザーごとのセッションの最大数を設定します。デフォルトは5分です。
- *number*：のセッションの数を設定します。その他のキーワードを指定せずに入力すると、この引数では 1 ～ 15 のセッションの集約数が設定されます。デフォルトは 15 です。

例：

```
ciscoasa(config)# quota management-session ssh 3
ciscoasa(config)# quota management-session telnet 1
ciscoasa(config)# quota management-session http 4
ciscoasa(config)# quota management-session user 2
```

ステップ 2 使用中の現在のセッションを表示します。

show quota management-session[**ssh** |**telnet** |**http** |**user**]

例：

```
ciscoasa(config)#show quota management-session

#Sessions               ConnectionType                 Username
1                             SSH                      cisco
2                             TELNET                   cisco
1                             SSH                      cisco1
```

システム管理者用 AAA の設定

この項では、システム管理者の認証、管理許可、コマンド許可を設定する方法について説明します。

管理認証の設定

CLI および ASDM アクセスの認証を設定します。

管理認証について

ASA へのログイン方法は、認証を有効にしているかどうかによって異なります。

SSH 認証の概要

認証ありまたは認証なしでの SSH アクセスについては、次の動作を参照してください。

- 認証なし：SSH は認証なしでは使用できません。
- 認証あり：SSH 認証を有効にした場合は、AAA サーバーまたはローカルユーザーデータベースに定義されているユーザー名とパスワードを入力します。公開キーの認証では、ASA はローカル データベースのみをサポートします。 SSH 公開キー認証を設定した場合、ASA ではローカル データベースを暗黙的に使用します。ログインにユーザー名とパスワードを使用する場合に必要なのは、SSH認証を明示的に設定することのみです。ユーザー EXEC モードにアクセスします。

Telnet 認証の概要

認証の有無にかかわらず、Telnet アクセスについては、次の動作を参照してください。

- 認証なし：Telnet の認証を有効にしていない場合は、ユーザー名を入力しません。ログイン パスワード（**password** コマンドで設定）を入力します。デフォルトのパスワードはありません。したがって、ASA へ Telnet 接続するには、パスワードを設定する必要があります。ユーザー EXEC モードにアクセスします。
- 認証あり：Telnet認証を有効にした場合は、AAAサーバーまたはローカルユーザーデータベースに定義されているユーザー名とパスワードを入力します。ユーザーEXECモードにアクセスします。

ASDM 認証の概要

認証ありまたは認証なしでのASDMアクセスに関しては、次の動作を参照してください。AAA認証の有無にかかわらず、証明書認証を設定することも可能です。

- 認証なし：デフォルトでは、ブランクのユーザー名と **enable password** コマンドによって設定されたイネーブル パスワード（デフォルトではブランク）を使用して ASDM にログインできます。空白のままにしないように、できるだけ早くイネーブルパスワードを変更

することをお勧めします。ホスト名、ドメイン名、およびイネーブル パスワードと Telnet パスワードの設定を参照してください。CLI で **enable** コマンドを最初に入力したときに、パスワードを変更するように求められます。ASDM にログインしたときには、この動作は適用されません。ログイン画面で（ユーザー名をブランクのままにしないで）ユーザー名とパスワードを入力した場合は、ASDM によってローカル データベースで一致がチェックされることに注意してください。

- 証明書認証（シングル、ルーテッド モードのみ）：ユーザーに有効な証明書を要求できます。証明書のユーザー名とパスワードを入力すると、ASA が PKI トラストポイントに対して証明書を検証します。
- AAA 認証：ASDM（HTTPS）認証を有効にした場合は、AAA サーバーまたはローカル ユーザー データベースに定義されているユーザー名とパスワードを入力します。これで、ブランクのユーザー名とイネーブル パスワードで ASDM を使用できなくなりました。
- AAA 認証と証明書認証の併用（シングル、ルーテッド モードのみ）：ASDM（HTTPS）認証を有効にした場合は、AAA サーバーまたはローカル ユーザー データベースに定義されているユーザー名とパスワードを入力します。証明書認証用のユーザー名とパスワードが異なる場合は、これらも入力するように求められます。ユーザー名を証明書から取得してあらかじめ入力しておくよう選択できます。

シリアル認証の概要

認証ありまたは認証なしでのシリアル コンソール ポートへのアクセスに関しては、次の動作を参照してください。

- 認証なし：シリアルアクセスの認証を有効にしていない場合は、ユーザー名、パスワードを入力しません。ユーザー EXEC モードにアクセスします。
- 認証あり：シリアルアクセスの認証を有効にした場合は、AAA サーバーまたはローカル ユーザー データベースで定義されているユーザー名とパスワードを入力します。ユーザー EXEC モードにアクセスします。

enable 認証の概要

ログイン後に特権 EXEC モードに入るには、**enable** コマンドを入力します。このコマンドの動作は、認証がイネーブルかどうかによって異なります。

- 認証なし：enable 認証を設定していない場合は、**enable** コマンドを入力するときにシステム イネーブル パスワード（**enable password** コマンドで設定）を入力します。デフォルトは空白です。**enable** コマンドを最初に入力したときに、それを変更するように求められます。ただし、enable 認証を使用しない場合、**enable** コマンドを入力した後は、特定のユーザーとしてログインしていません。これにより、コマンド認可などユーザーベースの各機能が影響を受けることがあります。ユーザー名を維持するには、enable 認証を使用してください。
- 認証あり：enable 認証を設定した場合は、ASA はプロンプトにより AAA サーバーまたはローカル ユーザー データベースで定義されているユーザー名とパスワードを要求します。

この機能は、ユーザーが入力できるコマンドを判別するためにユーザー名が重要な役割を果たすコマンド許可を実行する場合に特に役立ちます。

ローカル データベースを使用する enable 認証の場合は、**enable** コマンドの代わりに **login** コマンドを使用できます。**login** コマンドによりユーザー名が維持されますが、認証をオンにするための設定は必要ありません。

⚠

注意 CLI にアクセスできるユーザーや特権 EXEC モードを開始できないようにするユーザーをローカルデータベースに追加する場合は、コマンド認可を設定する必要があります。コマンド認可がない場合、特権レベルが 2 以上（2 がデフォルト）のユーザーは、CLI で自分のパスワードを使用して特権 EXEC モード（およびすべてのコマンド）にアクセスできます。あるいは、認証処理でローカルデータベースではなく AAA サーバーを使用してログイン コマンドを回避するか、またはすべてのローカル ユーザーをレベル 1 に設定することにより、システム イネーブル パスワードを使用して特権 EXEC モードにアクセスできるユーザーを制御できます。

ホスト オペレーティング システムから ASA へのセッション

一部のプラットフォームでは、ASA の実行を別のアプリケーションとしてサポートしています（例：Firepower 4100/9300 の ASA）。ホスト オペレーティング システムから ASA へのセッションの場合、接続のタイプに応じてシリアルおよび Telnet 認証を設定できます。

マルチ コンテキスト モードでは、システム コンフィギュレーションで AAA コマンドを設定できません。ただし、Telnet またはシリアル認証を管理コンテキストで設定した場合、認証はこれらのセッションにも適用されます。この場合、管理コンテキストの AAA サーバーまたはローカル ユーザー データベースが使用されます。

CLIおよびASDM アクセス認証の設定

始める前に

- Telnet、SSH、または HTTP アクセスを設定します。
- 外部認証の場合は、AAA サーバー グループを設定します。ローカル認証の場合は、ローカル データベースにユーザーを追加します。
- HTTP 管理認証では、AAA サーバー グループの SDI プロトコルをサポートしていません。
- この機能は、**ssh authentication** コマンドによるローカル ユーザー名に関する SSH 公開キー認証には影響しません。ASA では、公開キー認証に対し、ローカル データベースを暗黙的に使用します。この機能は、ユーザー名とパスワードにのみ影響します。ローカル ユーザーが公開キー認証またはパスワードを使用できるようにするには、この手順を使用してローカル認証を明示的に設定し、パスワード アクセスを許可する必要があります。

手順

管理アクセス用のユーザーを認証します。

aaa authentication {**telnet** | **ssh** | **http** | **serial**} **console** {**LOCAL** | *server_group* [**LOCAL**]}

例：

```
ciscoasa(config)# aaa authentication ssh console radius_1 LOCAL
ciscoasa(config)# aaa authentication http console radius_1 LOCAL
ciscoasa(config)# aaa authentication serial console LOCAL
```

telnet キーワードは Telnet アクセスを制御します。**ssh** キーワードは SSH アクセスを制御します（パスワードのみ。公開キー認証では暗黙のうちにローカルデータベースが使用されます）。**http** キーワードは ASDM アクセスを制御します。**serial** キーワードはコンソール ポート アクセスを制御します。

認証に AAA サーバー グループを使用する場合は、AAA サーバーが使用できないときにローカルデータベースをフォールバック方式として使用するように ASA を設定できます。サーバーグループ名を指定し、その後に **LOCAL**（大文字と小文字の区別あり）を追加します。ローカル データベースでは AAA サーバーと同じユーザー名およびパスワードを使用することを推奨します。これは、ASA のプロンプトでは、どの方式が使用されているかが示されないためです。**LOCAL** だけを入力して、ローカルデータベースを認証の主要方式として（フォールバックなしで）使用することもできます。

enable コマンド認証の設定（特権 EXEC モード）

ユーザーが **enable** コマンドを入力する際に、そのユーザーを認証できます。

始める前に

enable 認証の概要 （26 ページ）を参照してください。

手順

ユーザーを認証するための次のオプションのいずれかを選択します。

- AAA サーバーまたは LOCAL データベースを使用してユーザーを認証するには、次のコマンドを入力します。

  aaa authentication enable console {**LOCAL** | *server_group* [**LOCAL**]}

  例：

  ```
  ciscoasa(config)# aaa authentication enable console LOCAL
  ```

ユーザー名とパスワードの入力を求めるプロンプトがユーザーに対して表示されます。

認証に AAA サーバー グループを使用する場合は、AAA サーバーが使用できないときにローカルデータベースをフォールバック方式として使用するようにASAを設定できます。サーバー グループ名を指定し、その後に**LOCAL**（大文字と小文字の区別あり）を追加します。ローカル データベースではAAA サーバーと同じユーザー名およびパスワードを使用することを推奨します。これは、ASAのプロンプトでは、どの方式が使用されているかが示されないためです。

LOCALだけを入力して、ローカルデータベースを認証の主要方式として（フォールバックなしで）使用することもできます。

- ローカル データベースからユーザーとしてログインするには、次のコマンドを入力します。

  login

  例：

  ```
  ciscoasa# login
  ```

  ASA により、ユーザー名とパスワードの入力を求めるプロンプトが表示されます。パスワードを入力すると、ASA により、ユーザーはローカル データベースで指定されている特権レベルに置かれます。

  ユーザーは独自のユーザー名とパスワードでログインして特権EXECモードにアクセスすることができるので、システム イネーブル パスワードを全員に提供する必要がなくなります。ユーザーがログイン時に特権EXECモード（およびすべてのコマンド）にアクセスできるようにするには、ユーザーの特権レベルを 2（デフォルト）～ 15 に設定します。ローカル コマンド認可を設定した場合、ユーザーは、その特権レベル以下のレベルに割り当てられているコマンドのみを入力できます。

ASDM 証明書認証の構成

AAA 認証の有無にかかわらず証明書認証を必須にできます。ASA は証明書を PKI トラストポイントに照合して検証します。

始める前に

- シングルモードおよびルーテッドモードのみでサポートされています。
- クライアント証明書（.pfx 形式）およびサーバー証明書（.p12 形式）が必要です。証明書をインポートするための復号パスフレーズを記憶します。

手順

ステップ 1 ID 証明書を Cisco ASA に追加します。トラストポイントの設定を参照してください。

ステップ 2 証明書認証をイネーブルにします。

http authentication-certificate *interface_name*[**match** *certificate_map_name*]

例：

```
ciscoasa(config)# crypto ca certificate map map1 10
ciscoasa(config-ca-cert-map)# subject-name emailAddress www.example.com
ciscoasa(config)# http authentication-certificate outside match map1
```

証明書認証はインターフェイスごとに設定できます。その結果、信頼できるインターフェイスまたは内部インターフェイス上の接続については証明書の提示が不要になります。コマンドを複数回使用すれば、複数のインターフェイス上で証明書認証をイネーブルにできます。

証明書が証明書マップと一致することを要件にするには、**match** キーワードとマップ名を指定します。**crypto ca certificate map** コマンドを使用して、マップを設定します。

ステップ 3 （任意） ASDM で証明書からユーザー名を抽出する際に使用する属性を設定します。

http username-from-certificate{*primary-attr* [*secondary-attr*] | **use-entire-name** | **use-script**} [**pre-fill-username**]

例：

```
ciscoasa(config)# http username-from-certificate CN pre-fill-username
```

デフォルトでは、ASDM は CN OU 属性を使用します。

- *primary-attr* 引数は、ユーザ名の抽出に使用する属性を指定します。*secondary-attr* 引数は、オプションで、ユーザー名を抽出するためにプライマリ属性と一緒に使用する追加の属性を指定します。次の属性を使用できます。
  - C：国
  - CN：共通名
  - DNQ：DN 修飾子
  - emailAddress：電子メールアドレス
  - GENQ：世代修飾子
  - GN：名
  - I：イニシャル
  - L：局所性
  - N：名前
  - O：組織

- OU：組織単位
- SER：シリアル番号
- SN：姓
- SP：都道府県
- T：役職
- UID：ユーザー ID
- UPN：ユーザー プリンシパル名

- **use-entire-name** キーワードでは DN 名全体を使用します。
- **use-script** キーワードでは ASDM によって生成された Lua スクリプトを使用します。
- **pre-fill-username** キーワードでは、認証を求めるプロンプトにユーザー名が事前入力されています。そのユーザー名が最初に入力したものと異なる場合、最初のユーザー名が事前入力された新しいダイアログボックスが表示されます。そこに、認証用のパスワードを入力できます。

ステップ 4 Windows ASDM Launcher の手順。

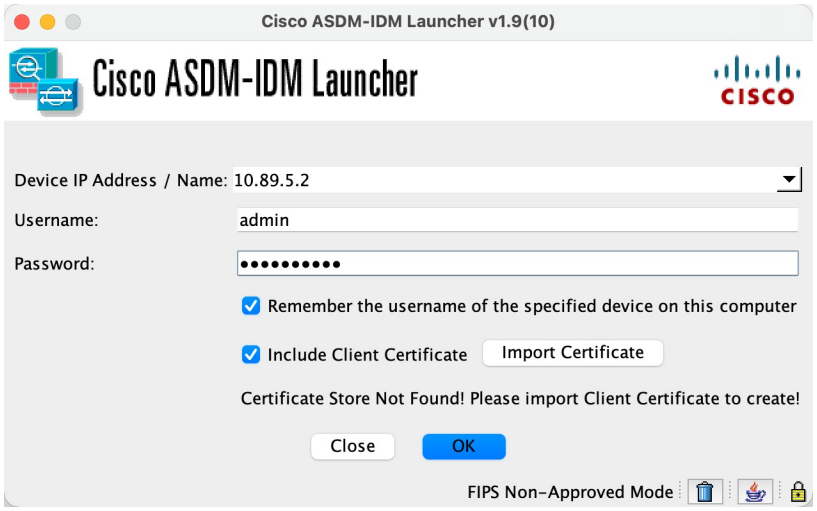
a) （FIPS 非承認モード）クライアント証明書をインストールします。

1. .pfx ファイルをダブルクリックして、証明書をインストールします。
2. 復号パスワードを入力します。

   Windows の証明書の詳細については、https://learn.microsoft.com/en-us/dotnet/framework/wcf/feature-details/working-with-certificates を参照してください。

b) ASDM Launcher を起動します。

図 1 : ASDM Launcher



c) [デバイスのIPアドレス（Device IP Address）]、[ユーザー名（Username）]および[パスワード（Password）]を入力します。
d) [クライアント証明書を含める（Include Client Certificate）]をオンにします。
e) （FIPS 承認モード）[証明書をインポート（Import Certificate）]をオンにして、.pfx 形式のクライアント証明書を参照します。パスワードを入力し、[OK] をクリックします。

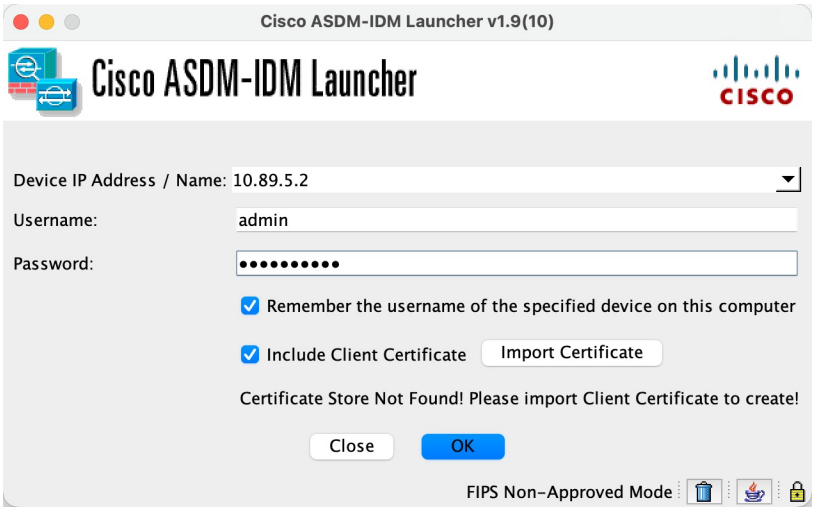
Windows での FIPS 非承認モードでは、クライアント証明書は、Windows 証明書ストアで取得できます。また、[証明書をインポート（Import Certificate）]ボタンが表示されます。

f) （FIPS 承認モード）証明書を更新するには、[証明書を更新（Update Certificate）]ボタンをクリックして、新しい証明書をインポートします。

ステップ 5 MacOS ASDM Launcher の手順。

a) ASDM Launcher を起動します。

図 2 : ASDM Launcher



b) [デバイスのIPアドレス（Device IP Address）]、[ユーザー名（Username）]および[パスワード（Password）]を入力します。
c) [クライアント証明書を含める（Include Client Certificate）]をオンにします。
d) [証明書をインポート（Import Certificate）]をオンにして、.pfx 形式のクライアント証明書を参照します。パスワードを入力し、[OK] をクリックします。
e) 証明書を更新するには、[証明書を更新（Update Certificate）]ボタンをクリックして、新しい証明書をインポートします。

管理許可による CLI および ASDM アクセスの制限

ASA ではユーザーの認証時に管理アクセス ユーザーとリモート アクセス ユーザーを区別できるようになっています。ユーザー ロールを区別することで、リモート アクセス VPN ユーザーやネットワーク アクセス ユーザーが ASA に管理接続を確立するのを防ぐことができます。

SSH X.509 証明書の承認については、「X.509 証明書アクセス用の SSH の設定 （11 ページ）」を参照してください。

始める前に

RADIUS または LDAP（マッピング済み）ユーザー

ユーザーが LDAP 経由で認証されると、ネイティブ LDAP 属性とその値が Cisco ASA 属性にマッピングされ、特定の許可機能が提供されます。Cisco VSA CVPN3000-Privilege-Level の値を 0 ～ 15 の範囲で設定した後、ldap map-attributes**ldap map-attributes** コマンドを使用して、LDAP 属性を Cisco VAS CVPN3000-Privilege-Level にマッピングします。

RADIUS IETF の **service-type** 属性が、RADIUS 認証および許可要求の結果として access-accept メッセージで送信される場合、この属性は認証されたユーザーにどのタイプのサービスを付与するかを指定するために使用されます。

RADIUS Cisco VSA **privilege-level** 属性（ベンダー ID 3076、サブ ID 220）が access-accept メッセージで送信される場合は、ユーザーの権限レベルを指定するために使用されます。

TACACS+ ユーザー

「service=shell」で許可が要求され、サーバーは PASS または FAIL で応答します。

ローカル ユーザー

指定したユーザー名に対する **service-type** コマンドを設定します。デフォルトでは、service-type は admin で、**aaa authentication console** コマンドで指定されたすべてのサービスに対してフルアクセスが許可されます。

管理許可の属性

管理許可の AAA サーバー タイプおよび有効な値については、次の表を参照してください。ASA ではこれらの値を使用して管理アクセス レベルを決定します。

管理レベル	RADIUS/LDAP の（マッピングされた）属性	TACACS+ 属性	ローカル データベースの属性
[Full Access]： **aaa authentication console** コマンド	Service-Type 6（アドミニストレーティブ）、Privilege-Level 1	PASS、特権レベル 1	admin
[Partial Access]： **aaa authentication console** コマンドで設定すると、CLI または ASDM に対するアクセスが許可されます。ただし、**aaa authentication enable console** コマンドを使用して **enable** 認証を設定する場合、CLI y ユーザーは **enable** コマンドを使用して特権 EXEC モードにアクセスすることはできません。	Service-Type 7（NAS プロンプト）、Privilege-Level 2 以上 Framed (2) および Login (1) サービス タイプは同様に扱われます。	PASS、特権レベル 2 以上	nas-prompt

管理レベル	RADIUS/LDAP の（マッピングされた）属性	TACACS+ 属性	ローカル データベースの属性
[No Access]：管理アクセスが拒否されます。ユーザーは **aaa authentication console** コマンドで指定されたいずれのサービスも使用できません（**serial** キーワードは除きます。つまり、シリアル アクセスは許可されます）。リモート アクセス（IPsec および SSL）ユーザーは、引き続き自身のリモートアクセスセッションを認証および終了できます。他のすべてのサービスタイプ（ボイス、ファクスなど）も同様に処理されます。	Service-Type 5（アウトバウンド）	FAIL	remote-access

その他のガイドライン

- シリアル コンソール アクセスは管理許可に含まれません。
- この機能を使用するには、管理アクセスにAAA認証も設定する必要があります。CLIおよびASDM アクセス認証の設定 （27 ページ）を参照してください。
- 外部認証を使用する場合は、この機能をイネーブルにする前に、AAA サーバー グループを設定しておく必要があります。
- HTTP 許可は、シングル ルーテッド モードでのみサポートされます。

手順

ステップ 1 Telnet と SSH の管理許可をイネーブルにします。

aaa authorization exec {**authentication-server** | **LOCAL**} [**auto-enable**]

auto-enable キーワードを使用して、十分な認証特権を持つ管理者が、ログインするときに特権 EXEC モードに自動的に入ることができます。

例：

```
ciscoasa(config)# aaa authentication ssh console RADIUS
ciscoasa(config)# aaa authorization exec authentication-server auto-enable
```

ステップ 2 HTTPS の管理許可をイネーブルにします（ASDM）。

aaa authorization http console {**authentication-server** | **LOCAL**}

例：

```
ciscoasa(config)# aaa authentication http console RADIUS
```

```
ciscoasa(config)# aaa authorization http console authentication-server
```

ステップ 3

例

次の例は、LDAP 属性マップを定義する方法を示しています。この例では、セキュリティ ポリシーによって、LDAP によって認証されているユーザーが、ユーザー レコードのフィールドまたはパラメータの title と company を、IETF-RADIUS service-type と privilege-level にそれぞれマップすることを指定しています。

```
ciscoasa(config)# ldap attribute-map admin-control
ciscoasa(config-ldap-attribute-map)# map-name title IETF-RADIUS-Service-Type
ciscoasa(config-ldap-attribute-map)# map-name company
```

次の例では、LDAP 属性マップを LDAP AAA サーバーに適用します。

```
ciscoasa(config)# aaa-server ldap-server (dmz1) host 10.20.30.1
ciscoasa(config-aaa-server-host)# ldap attribute-map admin-control
```

コマンド認可の設定

コマンドへのアクセスを制御する場合、ASA ではコマンド許可を設定でき、ユーザーが使用できるコマンドを決定できます。デフォルトでは、ログインするとユーザー EXEC モードにアクセスでき、最低限のコマンドだけが提供されます。**enable** コマンド（または、ローカル データベースを使用するときは **login** コマンド）を入力すると、特権 EXEC モードおよびコンフィギュレーション コマンドを含む高度なコマンドにアクセスできます。

次の 2 つのコマンド許可方式のいずれかを使用できます。

- ローカル特権レベル
- TACACS+ サーバー特権レベル

コマンド認可について

コマンド認可を有効にし、承認済みのユーザーにのみコマンド入力を許容することができます。

サポートされるコマンド認可方式

次の 2 つのコマンド許可方式のいずれかを使用できます。

- ローカル特権レベル：ASA でコマンド特権レベルを設定します。ローカル ユーザー、RADIUS ユーザー、または LDAP ユーザー（LDAP 属性を RADIUS 属性にマッピングする場合）を CLI アクセスについて認証する場合、ASA はそのユーザーをローカル データベー

ス、RADIUS、または LDAP サーバーで定義されている特権レベルに所属させます。ユーザーは、割り当てられた特権レベル以下のコマンドにアクセスできます。すべてのユーザーは、初めてログインするときに、ユーザー EXEC モード（レベル 0 または 1 のコマンド）にアクセスします。ユーザーは、特権 EXEC モード（レベル 2 以上のコマンド）にアクセスするために再び **enable** コマンドで認証するか、**login** コマンドでログイン（ローカル データベースに限る）できます。

（注） ローカル データベース内にユーザーが存在しなくても、また CLI 認証や **enable** 認証がない場合でも、ローカル コマンド許可を使用できます。代わりに、**enable** コマンドを入力するときにシステム イネーブル パスワードを入力すると、ASA によってレベル 15 に置かれます。次に、すべてのレベルのイネーブル パスワードを作成します。これにより、**enable** *n*（2 ～ 15）を入力したときに、ASA によってレベル *n* に置かれるようになります。これらのレベルは、ローカル コマンド許可を有効にするまで使用されません。

- TACACS+ サーバー特権レベル：TACACS+ サーバーで、ユーザーまたはグループが CLI アクセスについて認証した後で使用できるコマンドを設定します。CLI でユーザーが入力するすべてのコマンドは、TACACS+ サーバーで検証されます。

セキュリティ コンテキストとコマンド許可

AAA 設定はコンテキストごとに個別であり、コンテキスト間で共有されません。

コマンド許可を設定する場合は、各セキュリティ コンテキストを別々に設定する必要があります。この設定により、異なるセキュリティ コンテキストに対して異なるコマンド許可を実行できます。

セキュリティ コンテキストを切り替える場合、管理者は、ログイン時に指定したユーザー名で許可されるコマンドが新しいコンテキスト セッションでは異なる可能性があることや、新しいコンテキストではコマンド許可がまったく設定されていない可能性があることを念頭に置いてください。コマンド許可がセキュリティ コンテキストによって異なる場合があることを管理者が理解していないと、混乱が生じる可能性があります。

（注） システム実行スペースでは AAA コマンドがサポートされないため、システム実行スペースではコマンド許可を使用できません。

コマンド権限レベル

デフォルトでは、次のコマンドが特権レベル 0 に割り当てられます。その他のすべてのコマンドは特権レベル 15 に割り当てられます。

- **show checksum**
- **show curpriv**

- **enable**
- **help**
- **show history**
- **login**
- **logout**
- **pager**
- **show pager**
- **clear pager**
- **quit**
- **show version**

コンフィギュレーション モード コマンドを 15 より低いレベルに移動する場合は、configure コマンドも同じレベルに移動してください。このようにしないと、ユーザーはコンフィギュレーション モードに入ることができません。

ローカル コマンド許可の設定

ローカル コマンド許可を使用して、コマンドを 16 の特権レベル（0 ～ 15）の 1 つに割り当てることができます。デフォルトでは、各コマンドは特権レベル 0 または 15 に割り当てられます。各ユーザーを特定の特権レベルに定義でき、各ユーザーは割り当てられた特権レベル以下のコマンドを入力できます。ASA は、ローカル データベース、RADIUS サーバー、または LDAP サーバー（LDAP 属性を RADIUS 属性にマッピングする場合）に定義されているユーザー特権レベルをサポートしています。

手順

ステップ 1 特権レベルにコマンドを割り当てます。

privilege [**show** | **clear** | **cmd**] **level** *level* [**mode** {**enable** | **cmd**}] **command** *commnad*

例：

```
ciscoasa(config)# privilege show level 5 command filter
```

再割り当てする各コマンドに対してこのコマンドを繰り返します。

このコマンドのオプションは、次のとおりです。

- **show**|**clear**|**cmd**：これらのオプション キーワードを使用すると、コマンドの show、clear、または configure 形式に対してだけ特権を設定できます。コマンドの configure 形式は、通常、未修正コマンド（**show** または **clear** プレフィックスなしで）または **no** 形式として、コンフィギュレーションの変更を引き起こす形式です。これらのキーワードのいずれかを使用しない場合は、コマンドのすべての形式が影響を受けます。

- **level** *level*：0 ～ 15 の重大度。
- **mode** {**enable** | **configure**}：ユーザー EXEC モードまたは特権 EXEC モードおよびコンフィギュレーションモードでコマンドを入力することができ、そのコマンドが各モードで異なるアクションを実行する場合は、それらのモードの特権レベルを個別に設定することができます。
  - **enable**：ユーザー EXEC モードと特権 EXEC モードの両方を指定します。
  - **configure**：**configure terminal** コマンドを使用してアクセスされるコンフィギュレーション モードを指定します。
- **command** *command*：設定しているコマンド。設定できるのは、*main* コマンドの特権レベルだけです。たとえば、すべての **aaa** コマンドのレベルを設定できますが、**aaa authentication** コマンドと **aaa authorization** コマンドのレベルを個別に設定できません。

ステップ 2 （任意） コマンド認可のための AAA ユーザーを有効にします。このコマンドを入力しない場合、ASA は、ローカル データベース ユーザーの特権レベルだけをサポートし、他のタイプのユーザーをすべてデフォルトでレベル 15 に割り当てます。

aaa authorization exec authentication-server [auto-enable]

例：

```
ciscoasa(config)# aaa authorization exec authentication-server
```

さらに、このコマンドは管理認証を有効にします。管理許可による CLI および ASDM アクセスの制限 （32 ページ）を参照してください。

ステップ 3 ローカルのコマンド特権レベルの使用を有効にします。

aaa authorization command LOCAL

例：

```
ciscoasa(config)# aaa authorization command LOCAL
```

コマンド特権レベルを設定する場合は、このコマンドでコマンド許可を設定しない限り、コマンド許可は実行されません。

例

filter コマンドの形式は次のとおりです。

- **filter**（**configure** オプションにより表されます）
- **show running-config filter**

- **clear configure filter**

特権レベルを形式ごとに個別に設定することができます。または、このオプションを省略してすべての形式に同じ特権レベルを設定することもできます。次は、各形式を個別に設定する方法の例です。

```
ciscoasa(config)# privilege show level 5 command filter
ciscoasa(config)# privilege clear level 10 command filter
ciscoasa(config)# privilege cmd level 10 command filter
```

また、次の例では、すべての filter コマンドを同じレベルに設定する例を示します。

```
ciscoasa(config)# privilege level 5 command filter
```

show privilege コマンドは、形式を分けて表示します。

次の例では、**mode** キーワードの使用方法を示します。**enable** コマンドは、ユーザー EXEC モードから入力する必要があります。一方、**enable password** コマンドは、コンフィギュレーション モードでアクセスでき、最も高い特権レベルが必要です。

```
ciscoasa(config)# privilege cmd level 0 mode enable command enable
ciscoasa(config)# privilege cmd level 15 mode cmd command enable
ciscoasa(config)# privilege show level 15 mode cmd command enable
```

次の例では、**mode** キーワードを使用する追加コマンド（**configure** コマンド）を示します。

```
ciscoasa(config)# privilege show level 5 mode cmd command configure
ciscoasa(config)# privilege clear level 15 mode cmd command configure
ciscoasa(config)# privilege cmd level 15 mode cmd command configure
ciscoasa(config)# privilege cmd level 15 mode enable command configure
```

（注） この最後の行は、**configure terminal** コマンドに関する行です。

TACACS+ サーバーでのコマンドの設定

グループまたは個々のユーザーの共有プロファイル コンポーネントとしての Cisco Secure Access Control Server（ACS）TACACS+ サーバーでコマンドを設定できます。サードパーティの TACACS+ サーバーの場合は、コマンド許可サポートの詳細については、ご使用のサーバーのマニュアルを参照してください。

Cisco Secure ACS バージョン 3.1 でコマンドを設定する場合は、次のガイドラインを参照してください。

- ASA は、シェル コマンドとして許可するコマンドを送信し、TACACS+ サーバーでシェル コマンドとしてコマンドを設定します。

（注） Cisco Secure ACS には、「pix-shell」と呼ばれるコマンドタイプが含まれている場合があります。このタイプは ASA コマンド許可に使用しないでください。

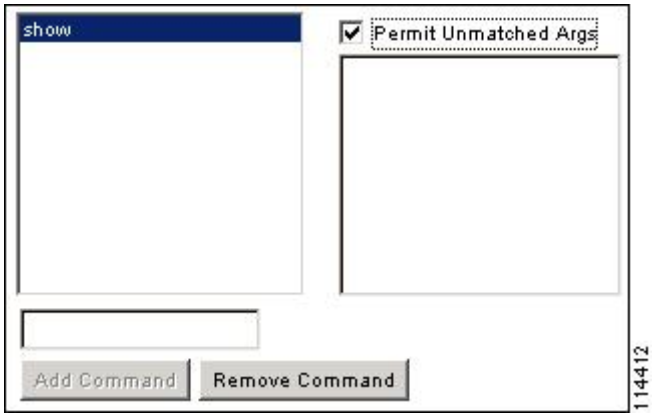
- コマンドの最初のワードは、メインコマンドと見なされます。その他のワードはすべて引数と見なされます。これは、**permit** または **deny** の後に置く必要があります。

  たとえば、**show running-configuration aaa-server** コマンドを許可するには、コマンドフィールドに **show running-configuration** を追加し、引数フィールドに **permit aaa-server** を入力します。

- [Permit Unmatched Args] チェックボックスをオンにすると、明示的に拒否していないすべてのコマンド引数を許可できます。

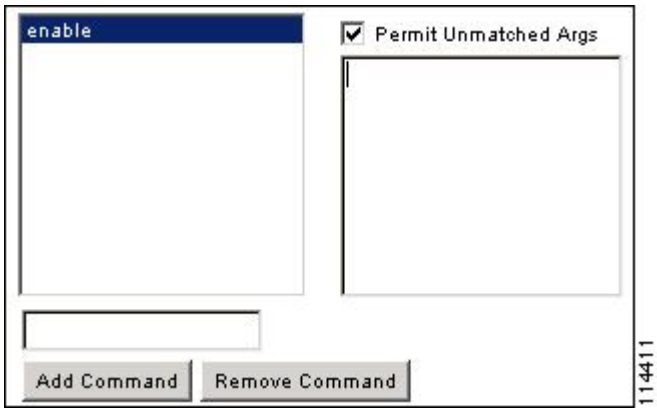
  たとえば、特定の **show** コマンドを設定するだけで、すべての **show** コマンドが許可されます。CLI の使用法を示す疑問符や省略形など、コマンドの変形をすべて予想する必要がなくなるので、この方法を使用することをお勧めします（次の図を参照）。

  図 *3*: 関連するすべてのコマンドの許可



- **enable** や **help** など、単一ワードのコマンドについては、そのコマンドに引数がない場合でも、一致しない引数を許可する必要があります（次の図を参照）。

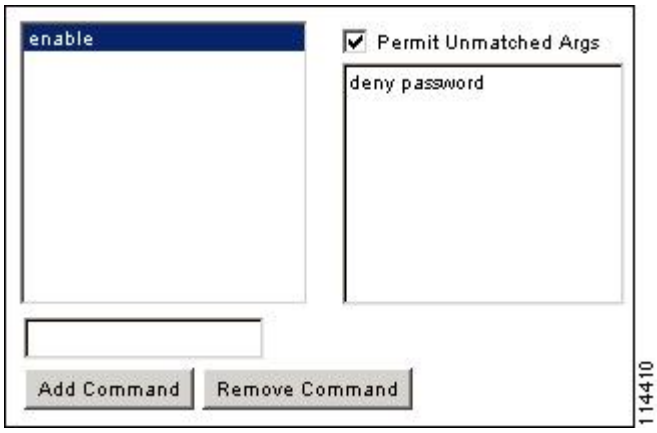
図 *4:* 単一ワードのコマンドの許可



- 引数を拒否するには、その引数の前に **deny** を入力します。

  たとえば、**enable** コマンドを許可し、**enable password** コマンドを許可しない場合には、コマンドフィールドに **enable** を入力し、引数フィールドに **deny password** を入力します。**enable** だけが許可されるように、必ず、[Permit Unmatched Args] チェックボックスをオンにしてください（次の図を参照）。

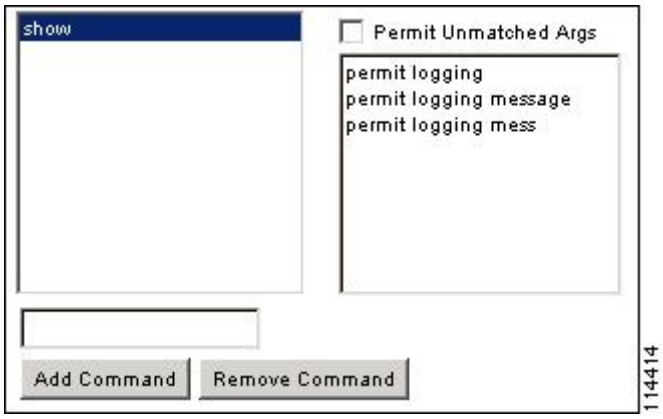
  図 *5:* 引数の拒否



- コマンドラインでコマンドを省略形で入力した場合、ASA はプレフィックスとメイン コマンドを完全なテキストに展開しますが、その他の引数は入力したとおりに TACACS+ サーバーに送信します。

  たとえば、**sh log** と入力すると、ASA は完全なコマンド **show logging** を TACACS+ サーバーに送信します。一方、**sh log mess** と入力すると、ASA は展開されたコマンド **show logging message** ではなく、**show logging mess** を TACACS+ サーバーに送信します。省略形を予想して同じ引数の複数のスペルを設定できます（次の図を参照）。

図 *6 :* 省略形の指定



- すべてのユーザーに対して次の基本コマンドを許可することをお勧めします。
  - **show checksum**
  - **show curpriv**
  - イネーブル化
  - **help**
  - **show history**
  - **login**
  - **logout**
  - **pager**
  - **show pager**
  - **clear pager**
  - **quit**
  - **show version**

TACACS+ コマンド許可の設定

TACACS+ コマンド認可をイネーブルにし、ユーザーが CLI でコマンドを入力すると、ASA はそのコマンドとユーザー名を TACACS+ サーバーに送信し、コマンドが認可されているかどうかを判別します。

TACACS+ コマンド許可をイネーブルにする前に、TACACS+ サーバーで定義されたユーザーとして ASA にログインしていること、および ASA の設定を続けるために必要なコマンド許可があることを確認してください。たとえば、すべてのコマンドが認可された管理ユーザーとしてログインする必要があります。このようにしないと、意図せずロックアウトされる可能性があります。

意図したとおりに機能することが確認できるまで、設定を保存しないでください。間違いによりロック アウトされた場合、通常はASAを再始動することによってアクセスを回復できます。

TACACS+ システムが完全に安定して信頼できることを確認します。必要な信頼性レベルについて、通常は、完全冗長 TACACS+ サーバー システムと ASA への完全冗長接続が必要です。たとえば、TACACS+ サーバー プールに、インターフェイス 1 に接続された 1 つのサーバーとインターフェイス 2 に接続された別のサーバーを含めます。TACACS+ サーバーが使用できない場合にフォールバック方式としてローカル コマンド許可を設定することもできます。

TACACS+ サーバーを使用したコマンド許可を設定するには、次の手順を実行します。

手順

次のコマンドを入力します。

aaa authorization command *tacacs+_server_group* [**LOCAL**]

例：

```
ciscoasa(config)# aaa authorization command tacacs+_server_group [LOCAL]
```

TACACS+サーバーを使用できない場合は、ローカル データベースをフォールバック方式として使用するように ASA を設定できます。フォールバックを有効にするには、サーバー グループ名の後ろに **LOCAL** を指定します（**LOCAL** は大文字と小文字を区別します）。ローカル データベースではTACACS+サーバーと同じユーザー名およびパスワードを使用することを推奨します。これは、ASAのプロンプトでは、どの方式が使用されているかが示されないためです。必ずローカル データベースのユーザーとコマンド特権レベルを設定してください。

ローカル データベース ユーザーのパスワード ポリシーの設定

ローカル データベースを使用して CLI または ASDM アクセスの認証を設定する場合は、指定期間を過ぎるとユーザーにパスワードの変更を要求し、パスワードの最短長と最低変更文字数などのパスワード標準に従うことを要求するパスワード ポリシーを設定できます。

パスワード ポリシーはローカル データベースを使用する管理ユーザーに対してのみ適用されます。ローカルデータベースを使用するその他のタイプのトラフィック（VPNやAAAによるネットワーク アクセスなど）や、AAA サーバーによって認証されたユーザーには適用されません。

パスワードポリシーの設定後は、自分または別のユーザーのパスワードを変更すると、新しいパスワードに対してパスワードポリシーが適用されます。既存のパスワードについては、現行のポリシーが適用されます。新しいポリシーは、 **username** コマンドおよび **change-password** コマンドを使用したパスワードの変更に適用されます。

始める前に

- ローカル データベースを使用して CLI または ASDM アクセスの AAA 認証を設定します。
- ローカル データベース内にユーザー名を指定します。

手順

ステップ 1 （オプション）リモート ユーザーのパスワードの有効期間を日数で設定します。

password-policy lifetime *days*

例：

```
ciscoasa(config)# password-policy lifetime 180
```

（注）
コンソール ポートを使用しているユーザーは、パスワードの有効期限が切れてもロックアウトされません。

有効な値は、0 ～ 65536 です。デフォルト値は 0 日です。この場合、パスワードは決して期限切れになりません。

パスワードの有効期限が切れる 7 日前に、警告メッセージが表示されます。パスワードの有効期限が切れると、リモート ユーザーのシステム アクセスは拒否されます。有効期限が切れた後アクセスするには、次のいずれかの手順を実行します。

- 他の管理者に **username** コマンドを使用してパスワードを変更してもらいます。
- 物理コンソール ポートにログインして、パスワードを変更します。

ステップ 2 （オプション）新しいパスワードと古いパスワードで違わなければならない最小文字数を設定します。

password-policy minimum-changes *value*

例：

```
ciscoasa(config)# password-policy minimum-changes 2
```

有効な値は、0 ～ 64 文字です。デフォルト値は 0 です

文字マッチングは位置に依存しません。したがって、新しいパスワードで使用される文字が、現在のパスワードのどこにも使用されていない場合に限り、パスワードが変更されたとみなされます。

ステップ 3 （オプション）パスワードの最小長を設定します。

password-policy minimum-length *value*

例：

```
ciscoasa(config)# password-policy minimum-length 8
```

有効な値は、3 ～ 64 文字です。推奨されるパスワードの最小長は 8 文字です。

ステップ 4 （オプション）パスワードに含める大文字の最小個数を設定します。

password-policy minimum-uppercase *value*

例：

```
ciscoasa(config)# password-policy  minimum-uppercase 3
```

有効な値は、0 ～ 64 文字です。デフォルト値は、最小個数がないことを意味する 0 です。

ステップ 5 （オプション）パスワードに含める小文字の最小個数を設定します。

password-policy minimum-lowercase *value*

例：

```
ciscoasa(config)# password-policy minimum-lowercase 6
```

有効な値は、0 ～ 64 文字です。デフォルト値は、最小個数がないことを意味する 0 です。

ステップ 6 （オプション）パスワードに含める数字の最小個数を設定します。

password-policy minimum-numeric *value*

例：

```
ciscoasa(config)# password-policy minimum-numeric 1
```

有効な値は、0 ～ 64 文字です。デフォルト値は、最小個数がないことを意味する 0 です。

ステップ 7 （オプション）パスワードに含める特殊文字の最小個数を設定します。

password-policy minimum-special *value*

例：

```
ciscoasa(config)# password-policy minimum-special 2
```

有効な値は、0 ～ 64 文字です。特殊文字には、!、@、#、$、%、^、&、*、(、および) が含まれます。デフォルト値は、最小個数がないことを意味する 0 です。

ステップ 8 パスワードを再利用を禁止します。

password-policy reuse-interval *value*

例：

```
ciscoasa(config)# password-policy reuse-interval 5
```

以前に使用された 2 ～ 7 個のパスワードと一致するパスワードの再利用を禁止することができます。以前のパスワードは、**password-history** コマンドを使用して、暗号化された形で各ユーザー名の設定に保存されます。このコマンドをユーザーが設定することはできません。

ステップ 9 ユーザー名と一致するパスワードを禁止します。

password-policy username-check

ステップ 10 （オプション）ユーザーが自分のパスワードの変更に **username** コマンドではなく **change-password** コマンドを使用する必要があるかを設定します。

password-policy authenticate enable

例：

```
ciscoasa(config)# password-policy authenticate enable
```

デフォルト設定はディセーブルです。どちらの方法でも、ユーザーはパスワードを変更することができます。

この機能を有効にして、**username** コマンドを使用してパスワードを変更しようとすると、次のエラー メッセージが表示されます。

```
ERROR: Changing your own password is prohibited
```

clear configure username コマンドを使用して自分のアカウントを削除することもできません。消去を試みた場合は、次のエラー メッセージが表示されます。

```
ERROR: You cannot delete all usernames because you are not allowed to delete yourself
```

パスワードの変更

パスワードポリシーでパスワードの有効期間を設定した場合、有効期間を過ぎるとパスワードを新しいパスワードに変更する必要があります。パスワードポリシー認証をイネーブルにした場合は、このパスワード変更のスキームが必須です。パスワードポリシー認証がイネーブルでない場合は、このメソッドを使用することも、直接ユーザーアカウントを変更することもできます。

username パスワードを変更するには、次の手順を実行します。

手順

次のコマンドを入力します。

change-password [**old-password** *old_password* [**new-password** *new_password*]]

例：

```
ciscoasa# change-password old-password j0hncr1chton new-password a3rynsun
```

コマンドに新旧のパスワードを入力していない場合は、ASA によって入力が求められます。

ログインの履歴を有効にして表示する

デフォルトでは、ログイン履歴は 90 日間保存されます。この機能を無効にするか、期間を最大 365 日まで変更できます。

始める前に

- ログイン履歴はユニット（装置）ごとに保存されます。フェールオーバーおよびクラスタリング環境では、各ユニットが自身のログイン履歴のみを保持します。
- ログインの履歴データは、リロードされると保持されなくなります。
- 1 つ以上の CLI 管理方式（SSH、Telnet、シリアル コンソール）でローカル AAA 認証をイネーブルにした場合、AAA サーバーのユーザー名またはローカルデータベースのユーザー名にこの機能が適用されます。ASDM のログインは履歴に保存されません。

手順

ステップ 1 ログインの履歴の期間を次のように設定します。

aaa authentication login-history duration *days*

例：

```
ciscoasa(config)# aaa authentication login-history duration 365
```

days を 1 ～ 365 日に設定できます。デフォルトは 90 です。ログイン履歴を無効にするには、**no aaa authentication login-history** を入力します。

ユーザーがログインすると、以下の SSH の例のように、自身のログイン履歴が表示されます。

```
cugel@10.86.194.108's password:
The privilege level for user cugel is 15.  The privilege level at the previous login was
 2.
User cugel logged in to ciscoasa at 21:04:10 UTC Dec 14 2016
Last login: 21:01:44 UTC Dec 14 2016 from ciscoasa console
Successful logins over the last 90 days: 6
Authentication failures since the last login: 0
Type help or '?' for a list of available commands.
ciscoasa>
```

ステップ 2 ログイン履歴を次のように表示します。

show aaa login-history [**user** *name*]

例：

```
ciscoasa(config)# show aaa login-history
Login history for user:    turjan
Logins in last   1 days:   1
Last successful login:     16:44:32 UTC Jul 23 2018 from console
Failures since last login: 0
Last failed login:         None
Privilege level:                                    14
Privilege level changed from 11 to 14 at:           14:07:30 UTC Aug 21 2018
```

管理アクセス アカウンティングの設定

CLIで**show**コマンド以外のコマンドを入力する場合、アカウンティング メッセージをTACACS+アカウンティング サーバーに送信できます。ユーザーがログインするとき、ユーザーが**enable**コマンドを入力するとき、またはユーザーがコマンドを発行するときのアカウンティングを設定できます。

コマンド アカウンティングに使用できるサーバーは、TACACS+ だけです。

管理アクセスおよびイネーブル コマンド アカウンティングを設定するには、次の手順を実行します。

手順

ステップ 1 次のコマンドを入力します。

aaa accounting {**serial** | **telnet** | **ssh** | **enable**} **console** *server-tag*

例：

```
ciscoasa(config)# aaa accounting telnet console group_1
```

有効なサーバー グループ プロトコルは RADIUS と TACACS+ です。

ステップ 2 コマンドアカウンティングをイネーブルにします。TACACS+サーバーだけがコマンドアカウンティングをサポートします。

aaa accounting command [**privilege** *level*] *server-tag*

例：

```
ciscoasa(config)# aaa accounting command privilege 15 group_1
```

privilege *level* というキーワードと引数のペアは最小特権レベルであり、*server-tag* 引数は ASA がコマンドアカウンティング メッセージを送信する TACACS+ サーバー グループの名前です。

ロックアウトからの回復

状況によっては、コマンド許可やCLI認証をオンにすると、ASA CLIからロックアウトされる場合があります。通常は、ASAを再起動することによってアクセスを回復できます。ただし、すでにコンフィギュレーションを保存した場合は、ロックアウトされたままになる可能性があります。

次の表に、一般的なロックアウト条件とその回復方法を示します。

表 *1 : CLI* 認証およびコマンド許可のロックアウト シナリオ

機能	ロックアウト条件	説明	対応策：シングル モード	対応策：マルチ モード
ローカル CLI 認証	ローカル データベースにユーザーが設定していない。	ローカル データベース内にユーザーが存在しない場合は、ログインできず、ユーザーの追加もできません。	ログインし、パスワードと **aaa** コマンドをリセットします。	スイッチから ASA へのセッションを接続します。システム実行スペースから、コンテキストに切り替えてユーザーを追加することができます。
TACACS+ コマンド許可 TACACS+ CLI 認証 RADIUS CLI 認証	サーバーがダウンしているか到達不能で、フォールバック方式を設定していない。	サーバーが到達不能である場合は、ログインもコマンドの入力もできません。	**1.** ログインし、パスワードと AAA コマンドをリセットします。 **2.** サーバーがダウンしたときにロックアウトされないように、ローカル データベースをフォールバック方式として設定します。	**1.** ASA でネットワーク コンフィギュレーションが正しくないためにサーバーが到達不能である場合は、スイッチから ASA へのセッションを接続します。システム実行スペースから、コンテキストに切り替えてネットワークを再設定することができます。 **2.** サーバーがダウンしたときにロックアウトされないように、ローカル データベースをフォールバック方式として設定します。

機能	ロックアウト条件	説明	対応策：シングル モード	対応策：マルチ モード
TACACS+ コマンド許可	十分な特権のないユーザーまたは存在しないユーザーとしてログインした。	コマンド許可がイネーブルになりますが、ユーザーはこれ以上コマンドを入力できなくなります。	TACACS+ サーバーのユーザーアカウントを修正します。 TACACS+ サーバーへのアクセス権がなく、ASA をすぐに設定する必要がある場合は、メンテナンス パーティションにログインして、パスワードと **aaa** コマンドをリセットします。	スイッチから ASA へのセッションを接続します。システム実行スペースから、コンテキストに切り替えてコンフィギュレーションの変更を完了することができます。また、TACACS+ コンフィギュレーションを修正するまでコマンド許可をディセーブルにすることもできます。
ローカル コマンド許可	十分な特権のないユーザーとしてログインしている。	コマンド許可がイネーブルになりますが、ユーザーはこれ以上コマンドを入力できなくなります。	ログインし、パスワードと **aaa** コマンドをリセットします。	スイッチから ASA へのセッションを接続します。システム実行スペースから、コンテキストに切り替えてユーザー レベルを変更することができます。

デバイス アクセスのモニタリング

デバイス アクセスのモニタリングについては、次のコマンドを参照してください。

- **show running-config all privilege all**

  このコマンドは、すべてのコマンドの特権レベルを表示します。

  show running-config all privilege all コマンドの場合、ASA は特権レベルに対する各 CLI コマンドの現在の割り当てを表示します。次に、このコマンドの出力例を示します。

```
ciscoasa(config)# show running-config all  privilege all
privilege show level 15 command aaa
privilege clear level 15 command aaa
privilege configure level 15 command aaa
privilege show level 15 command aaa-server
privilege clear level 15 command aaa-server
privilege configure level 15 command aaa-server
privilege show level 15 command access-group
privilege clear level 15 command access-group
privilege configure level 15 command access-group
privilege show level 15 command access-list
privilege clear level 15 command access-list
privilege configure level 15 command access-list
privilege show level 15 command activation-key
privilege configure level 15 command activation-key
...
```

- **show running-config privilege level** *level*

  このコマンドは、特定の特権レベルのコマンドを示します。level 引数は、0 ～ 15 の範囲の整数になります。

  次の例は、特権レベル 10 に対するコマンド割り当てを示しています。

```
ciscoasa(config)# show running-config all privilege level 10
privilege show level 10 command aaa
```

- **show running-config privilege command** コマンド

  このコマンドは、特定のコマンドの特権レベルを表示します。

  次の例は、**access-list** コマンドに対するコマンド割り当てを示しています。

```
ciscoasa(config)# show running-config all privilege command access-list
privilege show level 15 command access-list
privilege clear level 15 command access-list
privilege configure level 15 command access-list
```

- **show curpriv**

  このコマンドは、現在のログイン ユーザーを表示します。

  次に、**show curpriv** コマンドの出力例を示します。

```
ciscoasa# show curpriv
Username: admin
Current privilege level: 15
Current Mode/s: P_PRIV
```

次の表で、**show curpriv** コマンドの出力について説明します。

表 *2 : show curpriv* コマンド出力の説明

フィールド	説明
[ユーザー名（Username）]	[Username]。デフォルト ユーザーとしてログインすると、名前は enable_1（ユーザー EXEC）または enable_15（特権 EXEC）になります。
Current privilege level	レベルの範囲は 0 ～ 15 です。ローカル コマンド許可を設定してコマンドを中間特権レベルに割り当てない限り、使用されるレベルはレベル 0 と 15 だけです。
Current Modes	使用可能なアクセス モードは次のとおりです。 • P_UNPR：ユーザー EXEC モード（レベル 0 と 1） • P_PRIV：特権 EXEC モード（レベル 2 ～ 15） • P_CONF：コンフィギュレーション モード

- **show quota management-session** [**ssh** | **telnet** | **http** | **username** *user*]

  このコマンドは、使用中の現在のセッションを表示します。

  次に、**show quota management-session** コマンドの出力例を示します。

```
ciscoasa(config)#show quota management-session

#Sessions               ConnectionType                  Username
1                              SSH                      cisco
2                              TELNET                   cisco
1                              SSH                      cisco1
```

- **show aaa login-history** [**user** *name*]

  このコマンドは、ユーザーごとのログイン履歴を表示します。

  次に、**show aaa login-history** コマンドの出力例を示します。

```
ciscoasa(config)# show aaa login-history
Login history for user:    turjan
Logins in last   1 days:   1
Last successful login:     16:44:32 UTC Jul 23 2018 from console
Failures since last login: 0
Last failed login:         None
Privilege level:                                   14
Privilege level changed from 11 to 14 at:          14:07:30 UTC Aug 21 2018
```

管理アクセスの履歴

表 *3:* 管理アクセスの履歴

機能名	プラットフォームリリース	説明
SSH X.509 証明書認証	9.20(4)/9.24(1)	X.509v3 証明書を使用して SSH のユーザーを認証できるようになりました（RFC 6187）。 （注） この機能は、将来の FXOS リリースの Firepower 4100/9300 でサポートされる予定です。 （注） ASDM 7.20(4) のバンドルバージョンには、この機能のサポートは含まれていません。機能をサポートするには、Cisco.com から ASDM 7.20(4) をダウンロードしてインストールしてください。バンドルバージョンを上書きする場合は、必ずイメージ名を asdm.bin に変更してください。 新規/変更されたコマンド：**aaa authorization exec ssh-x509**、**ssh authentication method**、**ssh trustpoint sign**、**ssh username-from-certificate**、**validation-usage ssh-client**
ASDM 証明書認証	9.24(1)	ASDM 7.24 に付属している ASDM ランチャー 1.9(10) では、ユーザー証明書認証がサポートされるようになりました。以前は、この機能は Java Web Start でのみサポートされていました（7.18 で廃止）。ASA コマンドが 9.18 で廃止されていないため、ASDM ランチャー 1.9(10) を含む ASDM バージョンを使用する場合は証明書認証を使用するように以前の ASA バージョンを設定できます。 新規/変更されたコマンド：**http authentication-certificate**、**http username-from-certificate** 新規/変更された画面： • ASDM ランチャーのログインウィンドウ。
AES-256-GCM SSH 暗号	9.20(4)/9.24(1)	ASA は、SSH の AES-256-GCM 暗号をサポートしています。デフォルトでは、暗号化レベル [すべて（all）] と [高（high）] で有効になっています。 新規/変更されたコマンド： **ssh cipher encryption**

機能名	プラットフォームリリース	説明
Cisco ASA SSH スタックが廃止されました	9.23(1)	Cisco ASA SSH スタックを使用できなくなりました。Cisco SSH スタックが唯一のスタックになりました。 新規/変更されたコマンド： **ssh stack ciscossh**
CiscoSSH スタックのデフォルト化	9.19(1)	Cisco SSH スタックがデフォルトで使用されるようになりました。 新規/変更されたコマンド： **ssh stack ciscossh**
SSH と Telnet のループバック インターフェイス サポート	9.18(2)	ループバック インターフェイスを追加して、次の機能に使用できるようになりました。 • SSH • Telnet 新規/変更されたコマンド：**interface loopback**、**ssh**、**telnet**
CiscoSSH スタック	9.17(1)	ASA は、SSH 接続に独自の SSH スタックを使用します。代わりに、OpenSSH に基づく CiscoSSH スタックを使用するように選択できるようになりました。デフォルトスタックは引き続き ASA スタックです。Cisco SSH は次をサポートします。 • FIPS の準拠性 • シスコおよびオープンソースコミュニティからの更新を含む定期的な更新 CiscoSSH スタックは次をサポートしないことに注意してください。 • VPN を介した別のインターフェイスへの SSH（管理アクセス） • EdDSA キーペア • FIPS モードの RSA キーペア これらの機能が必要な場合は、引き続き ASA SSH スタックを使用する必要があります。 CiscoSSH スタックでは、SCP 機能に若干の変更があります。ASA **copy** コマンドを使用して SCP サーバとの間でファイルをコピーするには、**ssh** コマンドを使用して、ASA で SCP サーバサブネット/ホストの SSH アクセスを有効にする必要があります。 新規/変更されたコマンド： **ssh stack ciscossh**

機能名	プラットフォームリリース	説明
ローカルユーザーのロックアウトの変更	9.17(1)	設定可能な回数のログイン試行に失敗すると、ASA はローカルユーザーをロックアウトする場合があります。この機能は、特権レベル 15 のユーザーには適用されませんでした。また、管理者がアカウントのロックを解除するまで、ユーザーは無期限にロックアウトされます。管理者がその前に **clear aaa local user lockout** コマンドを使用しない限り、ユーザーは 10 分後にロック解除されるようになりました。特権レベル 15 のユーザーも、ロックアウト設定が適用されるようになりました。 新規/変更されたコマンド：**aaa local authentication attempts max-fail** 、**show aaa local user**
SSH および Telnet パスワード変更プロンプト	9.17(1)	ローカルユーザーが SSH または Telnet を使用して ASA に初めてログインすると、パスワードを変更するように求められます。また、管理者がパスワードを変更した後、最初のログインに対してもプロンプトが表示されます。ただし、ASA がリロードすると、最初のログインであっても、ユーザーにプロンプトは表示されません。 VPN などのローカル ユーザー データベースを使用するサービスは、SSH または Telnet ログイン中に変更された場合、新しいパスワードも使用する必要があることに注意してください。 新規/変更されたコマンド： **show aaa local user**

機能名	プラットフォームリリース	説明
SSH セキュリティの改善	9.16(1)	SSH が次の SSH セキュリティの改善をサポートするようになりました。 • ホストキーの形式：**crypto key generate** {**eddsa** | **ecdsa**}。RSA に加えて、EdDSA および ECDSA ホストキーのサポートが追加されました。ASA は、存在する場合、EdDSA、ECDSA、RSA の順にキーの使用を試みます。**ssh key-exchange hostkey rsa** コマンドで RSA キーを使用するように ASA を明示的に設定する場合は、2048 ビット以上のキーを生成する必要があります。アップグレードの互換性のために、ASA はデフォルトのホストキー設定が使用されている場合にのみ、より小さい RSA ホストキーを使用します。RSA のサポートは今後のリリースで削除されます。 • キー交換アルゴリズム：**ssh key-exchange group** {**ecdh-sha2-nistp256** | **curve25519-sha256**} • 暗号化アルゴリズム：**ssh cipher encryption chacha20-poly1305@openssh.com** • SSH バージョン 1 はサポートされなくなりました。**ssh version** コマンドは削除されました。 新規/変更されたコマンド：**crypto key generate eddsa**、**crypto key zeroize eddsa**、**show crypto key mypubkey**、**ssh cipher encryption chacha20-poly1305@openssh.com**、**ssh key-exchange group** {**ecdh-sha2-nistp256** | **curve25519-sha256**}、**ssh key-exchange hostkey**、**ssh version**
SNMP 向け管理アクセス	9.14(2)	サイト間 VPN 経由のセキュアな SNMP ポーリングを実現するための VPN 設定の一環として、VPN トンネル経由の管理アクセスを設定する際に、外部インターフェイスの IP アドレスを暗号マップアクセスリストに含めます。
HTTPS アイドルタイムアウトの設定	9.14(1)	ASDM、WebVPN、および他のクライアントを含む、ASA へのすべての HTTPS 接続のアイドルタイムアウトを設定できるようになりました。これまでは、**http server idle-timeout** コマンドを使用して ASDM アイドルタイムアウトを設定することしかできませんでした。両方のタイムアウトを設定した場合は、新しいコマンドによる設定が優先されます。 新規/変更されたコマンド： **http connection idle-timeout**

機能名	プラットフォームリリース	説明
事前定義されたリストに応じて最も高いセキュリティから最も低いセキュリティへという順序でSSH暗号化の暗号を表示	9.13(1)	事前定義されたリストに応じて、SSH暗号化の暗号が最も高いセキュリティから最も低いセキュリティへという順序（中または高）で表示されるようになりました。以前のリリースでは、最も低いものから最も高いものへの順序でリストされており、セキュリティが高い暗号よりも低い暗号が先に表示されていました。 新規/変更されたコマンド： **ssh cipher encryption**
SSHキー交換モードの設定は、管理コンテキストに限定されています。	9.12(2)	管理コンテキストではSSHキー交換を設定する必要があります。この設定は、他のすべてのコンテキストによって継承されます。 新規/変更されたコマンド： **ssh key-exchange**
enableログイン時のパスワードの変更が必須に	9.12(1)	デフォルトの**enable**のパスワードは空白です。ASAで特権EXECモードへのアクセスを試行する場合に、パスワードを3文字以上の値に変更することが必須となりました。空白のままにすることはできません。**no enable password** コマンドは現在サポートされていません。 CLIで**aaa authorization exec auto-enable**を有効にすると、**enable**コマンド、**login** コマンド（特権レベル2以上のユーザー）、またはSSH/Telnetセッションを使用して特権EXECモードにアクセスできます。これらの方法ではすべて、イネーブル パスワードを設定する必要があります。 このパスワード変更の要件は、ASDMのログインには適用されません。ASDMのデフォルトでは、ユーザー名を使用せず**enable**パスワードを使用してログインすることができます。 新規/変更されたコマンド： **enable password**
管理セッションの設定可能な制限	9.12(1)	集約、ユーザー単位、およびプロトコル単位の管理セッションの最大数を設定できます。これまでは、セッションの集約数しか設定できませんでした。この機能がコンソール セッションに影響を与えることはありません。マルチ コンテキスト モードではHTTPSセッションの数を設定することはできず、最大セッション数は5で固定されています。また、**quota management-session** コマンドはシステム コンフィギュレーションでは受け入れられず、代わりにコンテキスト コンフィギュレーションで使用できるようになっています。集約セッションの最大数が15になりました。0（無制限）または16以上に設定してアップグレードすると、値は15に変更されます。 新規/変更されたコマンド：**quota management-session**、**show quota management-session**

機能名	プラットフォームリリース	説明
管理権限レベルの変更通知	9.12(1)	有効なアクセス（**aaa authentication enable console**）を認証するか、または特権 EXEC への直接アクセス（**aaa authorization exec auto-enable**）を許可すると、前回のログイン以降に割り当てられたアクセス レベルが変更された場合に ASA からユーザーへ通知されるようになりました。 新規/変更されたコマンド： **show aaa login-history**
SSH によるセキュリティの強化	9.12(1)	次の SSH セキュリティの改善を参照してください。 • Diffie-Hellman Group 14 SHA256 キー交換のサポート。この設定がデフォルトになりました。以前のデフォルトは Group 1 SHA1 でした。 • HMAC-SHA256 整合性暗号のサポート。デフォルトは、高セキュリティの暗号セット（hmac-sha2-256 のみ）になりました。以前のデフォルトは中程度のセットでした。 新規/変更されたコマンド：**ssh cipher integrity** 、**ssh key-exchange group dh-group14-sha256**
非ブラウザベースの HTTPS クライアントによる ASA へのアクセスの許可	9.12(1)	非ブラウザベースの HTTPS クライアントが ASA 上の HTTPS サービスにアクセスできるようにすることができます。デフォルトでは、ASDM、CSM、および REST API が許可されています。 新規/変更されたコマンド： **http server basic-auth-client**
RSA キー ペアは 3072 ビット キーをサポートしています	9.9(2)	モジュラス サイズを 3072 に設定できるようになりました。 新規または変更されたコマンド：**crypto key generate rsa modulus**
ブリッジ型仮想インターフェイス（BVI）の VPN 管理アクセス	9.9(2)	VPN の **management-access** がその BVI で有効になっている場合、**telnet**、**http**、**ssh** などの管理サービスを BVI で有効にできるようになりました。非 VPN 管理アクセスの場合は、ブリッジ グループ メンバ インターフェイスでこれらのサービスの設定を続行する必要があります。 新規または変更されたコマンド：**https**、**telnet**、**ssh**、**management-access**
SSH バージョン 1 の廃止	9.9(1)	SSH バージョン 1 は廃止され、今後のリリースで削除される予定です。デフォルト設定が SSH v1 と v2 の両方から SSH v2 のみに変更されました。 新規/変更されたコマンド： **ssh version**

機能名	プラットフォームリリース	説明
SSH公開キー認証を使用するユーザーの認証とパスワードを使用するユーザーの認証を区別します。	9.6(3)/9.8(1)	9.6(2) より前のリリースでは、ローカル ユーザー データベース（**ssh authentication**）を使用して AAA SSH 認証を明示的に有効にしなくても、SSH 公開キー認証（**aaa authentication ssh console LOCAL**）を有効にすることができました。9.6(2) では、ASA で AAA SSH 認証を明示的に有効にする必要がありました。このリリースでは、AAA SSH 認証を明示的に有効にする必要はありません。ユーザーに対して **ssh authentication** コマンドを設定すると、このタイプの認証を使用するユーザーのローカル認証がデフォルトで有効になります。さらに、明示的に AAA SSH 認証を設定すると、この設定はパスワード付きのユーザー名にのみ適用されます。また、任意の AAA サーバー タイプ（**aaa authentication ssh console radius_1** など）を使用できます。たとえば、一部のユーザーはローカル データベースを使用して公開キー認証を使用し、他のユーザーは RADIUS でパスワードを使用できます。 変更されたコマンドはありません。
ログイン履歴	9.8(1)	デフォルトでは、ログイン履歴は 90 日間保存されます。この機能を無効にするか、期間を最大 365 日まで変更できます。1 つ以上の管理メソッド（SSH、ASDM、Telnet など）でローカル AAA 認証を有効にしている場合、この機能はローカル データベースのユーザー名にのみ適用されます。 次のコマンドが導入されました。**aaa authentication login-history**、**show aaa login-history**
パスワードの再利用とユーザー名と一致するパスワードの使用を禁止するパスワード ポリシーの適用	9.8(1)	最大 7 世代にわたるパスワードの再利用と、ユーザー名と一致するパスワードの使用を禁止できるようになりました。 次のコマンドが導入されました。**password-history**、**password-policy reuse-interval**、**password-policy username-check**
ASDM に対する ASA SSL サーバー モード マッチング	9.6(2)	証明書マップと照合するために、証明書で認証を行う ASDM ユーザーに対して証明書を要求できるようになりました。 次のコマンドを変更しました。**http authentication-certificate match**

機能名	プラットフォームリリース	説明
SSH 公開キー認証の改善	9.6(2)	以前のリリースでは、ローカル ユーザー データベース（**aaa authentication ssh console LOCAL**）を使用して AAA SSH 認証を有効にしなくても、SSH 公開キー認証（**ssh authentication**）を有効にすることができました。この設定は修正されたため、AAA SSH認証を明示的に有効にする必要があります。ユーザーが秘密キーの代わりにパスワードを使用できないよう、パスワード未定義のユーザー名を作成できるようになりました。 次のコマンドが変更されました。**ssh authentication**、**username**
ASDM 管理認証	9.4(1)	HTTP アクセスと Telnet および SSH アクセス別に管理認証を設定できるようになりました。 次のコマンドが導入されました。 **aaa authorization http console**
証明書コンフィギュレーションの ASDM ユーザー名	9.4(1)	ASDM の証明書認証（**http authentication-certificate**）を有効にすると、ASDM が証明書からユーザー名を抽出する方法を設定できます。また、ログイン プロンプトでユーザー名を事前に入力して表示できます。 次のコマンドが導入されました。**http username-from-certificate**
改善されたワンタイムパスワード認証	9.2(1)	十分な認可特権を持つ管理者は、認証クレデンシャルを一度入力すると特権 EXEC モードに移行できます。**auto-enable** オプションが **aaa authorization exec** コマンドに追加されました。 次のコマンドが変更されました。**aaa authorization exec**。
HTTP リダイレクトの IPv6 サポート	9.1(7)/9.6(1)	ASDM アクセスまたはクライアントレス SSL VPN 用の HTTPS に HTTP リダイレクトを有効にすると、IPv6 アドレスへ送信されるトラフィックもリダイレクトできるようになりました。 次のコマンドに機能が追加されました。**http redirect**

機能名	プラットフォームリリース	説明
設定可能なSSH暗号機能と整合性アルゴリズム	9.1(7)/9.4(3)/9.5(3)/9.6(1)	ユーザーはSSH暗号化を管理するときに暗号化モードを選択し、さまざまなキー交換アルゴリズムに対して HMAC と暗号化を設定できます。アプリケーションに応じて、暗号の強度を強くしたり弱くする必要がある場合があります。セキュアなコピーのパフォーマンスは暗号化アルゴリズムに一部依存します。デフォルトで、ASA は 3des-cbc aes128-cbc aes192-cbc aes256-cbc aes128-ctr aes192-ctr aes256-ctr の順にアルゴリズムをネゴシエートします。提示された最初のアルゴリズム（3des-cbc）が選択された場合、aes128-cbc などの一層効率的なアルゴリズムが選択された場合よりも大幅にパフォーマンスが低下します。たとえば、提示された暗号方式に変更するには、**ssh cipher encryption custom aes128-cbc** を使用します。 次のコマンドが導入されました。**ssh cipher encryption**、**ssh cipher integrity**。
SSH の AES-CTR 暗号化	9.1(2)	ASA での SSH サーバーの実装が、AES-CTR モードの暗号化をサポートするようになりました。
SSH キー再生成間隔の改善	9.1(2)	SSH 接続は、接続時間 60 分間またはデータ トラフィック 1 GB ごとに再生成されます。 次のコマンドが導入されました。**show ssh sessions detail**。
マルチ コンテキスト モードの ASASM において、スイッチからの Telnet 認証および仮想コンソール認証をサポートしました。	8.5(1)	マルチ コンテキスト モードのスイッチから ASASM への接続はシステム実行スペースに接続しますが、これらの接続を制御するために管理コンテキストでの認証を設定できます。
ローカルデータベースを使用する場合の管理者パスワードポリシーのサポート	8.4(4.1)、9.1(2)	ローカル データベースを使用して CLI または ASDM アクセスの認証を設定する場合は、指定期間を過ぎるとユーザーにパスワードの変更を要求し、パスワードの最短長と最低変更文字数などのパスワード標準に従うことを要求するパスワード ポリシーを設定できます。 次のコマンドが導入されました。**change-password**、**password-policy lifetime**、**password-policy minimum changes**、**password-policy minimum-length**、**password-policy minimum-lowercase**、**password-policy minimum-uppercase**、**password-policy minimum-numeric**、**password-policy minimum-special**、**password-policy authenticate enable**、**clear configure password-policy**、**show running-config password-policy**.

機能名	プラットフォームリリース	説明
SSH 公開キー認証のサポート	8.4(4.1)、9.1(2)	ASA への SSH 接続の公開キー認証は、ユーザー単位で有効にできます。公開キーファイル（PKF）でフォーマットされたキーまたは Base64 キーを指定できます。PKF キーは、4096 ビットまで使用できます。ASA がサポートする Base64 形式（最大 2048 ビット）では大きすぎるキーについては、PKF 形式を使用します。 次のコマンドが導入されました。**ssh authenticaion**。 *PKF* キー形式のサポートは *9.1(2)* 以降のみです。
SSH キー交換の Diffie-Hellman グループ 14 のサポート	8.4(4.1)、9.1(2)	SSH キー交換に Diffie-Hellman グループ 14 が追加されました。これまでは、グループ 1 だけがサポートされていました。 次のコマンドが導入されました。**ssh key-exchange**。
管理セッションの最大数のサポート	8.4(4.1)、9.1(2)	同時 ASDM、SSH、Telnet セッションの最大数を設定できます。 次のコマンドが導入されました。**quota management-session**、**show running-config quota management-session**、**show quota management-session**。
SSH セキュリティが向上し、SSH デフォルトユーザー名はサポートされなくなりました。	8.4(2)	8.4(2) 以降、pix または asa ユーザー名とログイン パスワードで SSH を使用して ASA に接続することができなくなりました。SSH を使用するには、**aaa authentication ssh console LOCAL** コマンド（CLI）または [Configuration] > [Device Management] > [Users/AAA] > [AAA Access] > [Authentication (ASDM)] を使用して AAA 認証を設定してから、ローカルユーザーを定義する必要があります。定義するには、**username** コマンド（CLI）を入力するか、[Configuration] > [Device Management] > [Users/AAA] > [User Accounts (ASDM)] を選択します。ローカル データベースの代わりに AAA サーバーを認証に使用する場合、ローカル認証もバックアップの手段として設定しておくことをお勧めします。

機能名	プラットフォームリリース	説明
管理アクセス	7.0(1)	この機能が導入されました。 次のコマンドを導入しました。 **show running-config all privilege all**、**show running-config privilege level**、**show running-config privilege command**、**telnet**、**telnet timeout**、**ssh**、**ssh timeout**、**http**、**http server enable**、**asdm image disk**、**banner**、**console timeout**、**icmp**、**ipv6 icmp**、**management access**、**aaa authentication console**、**aaa authentication enable console**、**aaa authentication telnet | ssh console**、**service-type**、**login**、**privilege**、**aaa authentication exec authentication-server**、**aaa authentication command LOCAL**、**aaa accounting serial | telnet | ssh | enable console**、**show curpriv**、**aaa accounting command privilege**。

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。

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。

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。

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。