SSH および Telnet の設定

この章では、Cisco NX-OS デバイス上でセキュア シェル（SSH）プロトコルおよび Telnet を設定する手順について説明します。

この章は、次の項で構成されています。

SSH および Telnet について

ここでは、SSH および Telnet について説明します。

SSH サーバー

SSH サーバを使用すると、SSH クライアントは Cisco NX-OS デバイスとの間でセキュアな暗号化された接続を確立できます。SSH は強化暗号化を使用して認証を行います。Cisco NX-OS ソフトウェアの SSH サーバは、市販の一般的な SSH クライアントと相互運用ができます。

SSH がサポートするユーザ認証メカニズムには、Remote Authentication Dial-In User Service（RADIUS）、TACACS+、LDAP、およびローカルに格納されたユーザ名とパスワードを使用した認証があります。

SSH クライアント

SSH クライアントは、SSH プロトコルで稼働しデバイス認証および暗号化を提供するアプリケーションです。Cisco NX-OS デバイスは、SSH クライアントを使用して、別の Cisco NX-OS デバイスまたは SSH サーバの稼働する他のデバイスとの間で暗号化された安全な接続を確立できます。この接続は、暗号化されたアウトバウンド接続を実現します。認証と暗号化により、SSH クライアントは、セキュリティ保護されていないネットワーク上でもセキュアな通信を実現できます。

Cisco NX-OS ソフトウェアの SSH クライアントは、無償あるいは商用の SSH サーバと連係して動作します。

SSH サーバ キー

SSH では、Cisco NX-OS とのセキュアな通信を行うためにサーバ キーが必要です。SSH サーバ キーは、次の SSH オプションに使用できます。

- Rivest, Shamir, and Adelman（RSA）公開キー暗号化を使用した SSH バージョン 2
- Digital System Algrorithm（DSA）を使用した SSH バージョン 2
- 楕円曲線デジタル署名アルゴリズム（ECDSA）を使用した SSH バージョン 2

SSH サービスをイネーブルにする前に、適切なバージョンの SSH サーバ キー ペアを取得してください。使用中の SSH クライアント バージョンに応じて、SSH サーバ キー ペアを生成します。SSH サービスでは、SSH バージョン 2 に対応する以下の 2 通りのキー ペアを使用できます。

- **dsa** オプションでは、SSH バージョン 2 プロトコル用の DSA キー ペアを作成します。
- **rsa** オプションでは、SSH バージョン 2 プロトコル用の RSA キー ペアを作成します。
- **ecdsa** オプションでは、SSH バージョン 2 プロトコル用の ECDSA キー ペアを作成します。

デフォルトでは、Cisco NX-OS ソフトウェアは 1024 ビットの RSA キーを生成します。

SSH は、次の公開キー形式をサポートします。

- OpenSSH
- IETF SSH（SECSH）
- Privacy-Enhanced Mail（PEM）の公開キー証明書

Caution SSH キーをすべて削除すると、SSH サービスを開始できません。

デジタル証明書を使用した SSH 認証

Cisco NX-OS デバイスでの SSH 認証では、ホスト認証用に X.509 デジタル証明書をサポートしています。X.509 デジタル証明書は、メッセージの出所と整合性を保証するデータ項目です。これには安全な通信のための暗号化されたキーが含まれています。また、発信者のアイデンティティを証明するために信頼できる認証局（CA）によって署名されています。X.509 デジタル証明書のサポートにより、認証に DSA と RSA のいずれかのアルゴリズムを使用します。

証明書のインフラストラクチャでは、Secure Socket Layer（SSL）に対応し、セキュリティインフラストラクチャによってクエリーまたは通知を通じて最初に返される証明書が使用されます。証明書が信頼できる CA のいずれかで設定されており、無効にされたり期限が切れたりしていなければ、証明書の検証は成功します。

X.509 証明書を使用する SSH 認証用にデバイスを設定できます。認証に失敗した場合は、パスワードの入力が求められます。

X.509v3 証明書（RFC 6187）を使用する SSH 認証を設定できます。X.509v3 証明書ベースの SSH 認証では、スマートカードと組み合わせた証明書を使用して、シスコ デバイスへのアクセスの 2 要素認証を有効にします。SSH クライアントは、シスコパートナーの Pragma Systems によって提供されます。

ホスト アイデンティティに基づく認証（HIBA）を使用した SSH 認証

ホストベース認証は、サーバーの `known_hosts` ファイルでクライアントのホスト公開キーを確認することにより、クライアントのホストをサーバー（Cisco Nexus 9000 スイッチ）に対して認証するSSH認証方式です。これは、ユーザーまたはホストを認証ために認証局（CA）によって署名されたデジタル証明書を使用する SSH 証明書ベースの認証とは異なります。

ホスト アイデンティティ ベースの認証（HIBA）は、証明書にホスト承認情報を埋め込むことによって SSH 承認管理を一元化する方式です。

- ホスト承認情報は、ホスト証明書に組み込まれています。
- ユーザー証明書には、アクセス許可を指定する「付与」が含まれています。
- 認証は、認証局（CA）によって一元的に管理されます。

HIBA は SSH アクセス制御を簡素化し、管理オーバーヘッドを削減し、承認のための外部 AAA サーバーへの依存を排除します。

HIBA の利点

HIBA には、従来の SSH キー管理に比べて次のような利点があります。

HIBA の主なメリットは以下のとおりです：

- **管理の簡素化**：証明書ベースのアイデンティティによる一元化された承認により、管理が簡素化されます。
- **拡張性**： 大規模で複雑な環境での SSH アクセスの管理が簡素化されます。

- **依存関係の軽減**： 承認に関する外部 AAA サーバーへの依存性を排除し、最終手段としてのアクセスに適したものにします。
- **セキュリティの強化**：短期間の証明書を使用した一時的なアクセスと特権アクセスの制御を向上させます。

HIBA によるSSH 認証の仕組み

このプロセスでは、HIBA が構成されている場合に SSH 認証がどのように行われるかについて説明します。

process_summary

SSH サーバーは、HIBA 承認モジュールを呼び出し、認証中にユーザー証明書を処理します。アクセスは、構成されたホストアイデンティティと付与に対して HIBA モジュールがユーザーの証明書を正常に検証した場合に付与されます。HIBA 検証が失敗した場合、 SSH サーバーは、構成に応じて、他の認証方法にフォールバックする場合があります。

process_workflow

次の段階で、HIBA を使用した SSH 認証プロセスについて説明します。

1. **[SSH 接続試行（SSH Connection Attempt）]**：ユーザーは、 SSH でスイッチへの接続を試行します。
2. **[証明書の提示（Certificate Presentation）]**： SSH クライアントは、スイッチ上の SSH サーバーにユーザーの証明書を提示します。
3. **[HIBA モジュール呼び出し（HIBA Module Invocation）]**： SSH サーバーは、その構成（AuthorizedPrincipalsCommand）に基づいて、HIBA 承認モジュールを呼び出します。
4. **[証明書の検証（Certificate Validation）]**：HIBA モジュールは、次の検証を実行します：
   - 構成された HIBA CA と 照合してユーザー証明書の署名を確認します。
   - ホスト証明書からホスト アイデンティティを抽出します。
   - ホストアイデンティティと一致するユーザー証明書内の有効な「付与」をチェックします。
5. **[アクセス決定（Access Decision）]**：HIBA モジュールの検証に基づいて、次のいずれかが行われます：

属性...	結合できるフィールド	次の操作	結合できるフィールド
ユーザー証明書が HIBA モジュールによって正常に検証されました。	ターゲット ホストの有効な付与がユーザー証明書にあります。	ユーザーにアクセス権が付与されます。	SSH セッションが続行されます。

属性...	結合できるフィールド	次の操作	結合できるフィールド
ユーザー証明書が無効であるか、検証できません。	ユーザー証明書に有効な付与が見つかりませんでした。	HIBA モジュールによってアクセスが拒否されました。	SSH サーバーは、他の認証方法にフォールバックする場合があります（構成されている場合）。

SSH 認証の HIBA の構成

この手順では、 SSH ホスト アイデンティティ ベースの認可（HIBA）の構成について説明します。

この構成では、 SSH サーバー キーの生成、HIBA CA のトラストポイントの構成、 SSH ホスト証明書の登録、認証に HIBA を使用するための SSH サーバーの構成を行います。

（注） 初めて HIBA を構成する場合、ローカル ユーザー アカウントやその他の構成済み AAA サーバーなど、従来の SSH 認証方式を使用してスイッチにログインできます。HIBA を有効にしても、既存のローカル SSH ユーザーは、明示的にアカウントを削除しない限り、削除またはブロックされません。

始める前に

HIBA を構成する前に、次の点を確認してください：

- 認証局（CA）を含む、機能する PKI インフラストラクチャ。
- CA サーバーへの接続。

手順

ステップ 1 **configure terminal**

例：

```
switch# configure terminal
```

グローバル構成モードを開始します。

ステップ 2 **ssh key ecdsa** *bits*

例：

```
switch(config)# ssh key ecdsa 384
```

スイッチの ECDSA キーペアを生成します。この例では、384 ビットの ECDSA キーが使用されています。セキュリティ ポリシーとプラットフォームでサポートされるキー サイズを使用します。

ステップ 3 **ssh key export** *bootflash:file_name* **ecdsa**

例：

```
switch(config)# ssh key export bootflash:host_key ecdsa
            Enter Passphrase:
```

SSH ホスト ECDSA キーをブートフラッシュにエクスポートします。必要に合わせて `file_name` を置き換えます。

エクスポート後、SFTP を使用して `host_key` および `host_key.pub` ファイルをCAマシンに転送します：

```
switch(config)# feature sftp-server
# On CA machine:
sftp admin@<switch_ip>
sftp> get host_key .
sftp> get host_key.pub .
```

ステップ 4 **crypto ca trustpoint** *openssh-ca* **type ssh**

例：

```
switch(config)# crypto ca trustpoint openssh-ca type ssh
```

HIBA CA のトラストポイントを作成します。一貫性を保つために、 **openssh-ca** という名前を使用します。

ステップ 5 **crypto ca authenticate** *openssh-ca* **type ssh** *ecdsa-sha2-nistp384 public_key*

例：

```
switch(config-trustpoint)# crypto ca authenticate openssh-ca type ssh ecdsa-sha2-nistp384
 AAAAE2VjZHNhLXNoYTItbmlzdHAzODQAAAAIbmlzdHAzODQAAABhBBPiMs3fwftVUoMT...
/home/admin/.hiba-ca CA
```

CA 公開キーをインポートして HIBA CAを認証します。キー文字列を実際の CA 公開キーに置き換えます。

ステップ 6 **crypto ca enroll** *openssh-ca* **type ssh host-certificate** *ecdsa-sha2-nistp384-cert-v01@openssh.com certificate_content*

例：

```
switch(config)# crypto ca enroll openssh-ca type ssh host-certificate
ecdsa-sha2-nistp384-cert-v01@openssh.com
[redacted]
 root@switch
```

CA によって署名された SSH ホスト証明書を登録します。Google HIBA CA Wiki の手順に従って生成された証明書のコンテンツを使用します。

設定例： Linux での HIBA SSH クライアント

重要 次に、 Linuxシステムで HIBA SSH クライアントを設定するための **例** として、次の手順を示します。正確な手順と出力は、クライアント オペレーティング システムと SSH のバージョンによって異なる場合があります。確実な手順については、システムの公式な SSH ドキュメントを参照してください。

この手順では、SSH でホスト アイデンティティ ベースの認証（HIBA）を使用するためのクライアント側の設定について説明します。

（注） 「HIBA サーバー」という用語は、 Cisco Nexus 9000 スイッチで実行され、HIBA を使用するように設定された SSH サーバーを指します。

始める前に

HIBA SSH クライアントを設定する前に、次を確認してください。

- ホストに `openssh-client` が有効にインストールされていること。
- CA 公開キー（`ca.pub`）。
- ユーザー秘密キーおよび有効な HIBA 拡張との一致証明書。
- ユーザー公開キー（`key_rsa.pub` または同等のもの）。

手順

ステップ 1 **$ cat /etc/ssh/ssh_config**

例：

```
$ cat /etc/ssh/ssh_config
# Enable host key checking
StrictHostKeyChecking yes
# Declare our trusted CA
GlobalKnownHostsFile /etc/ssh/known_hosts
```

SSH クライアント設定の構成

`/etc/ssh/ssh_config` を編集して、厳密なホスト キーのチェックを有効にし、SSH 証明書の検証用の CA 公開キーを含む `GlobalKnownHostsFile` を指定します。

ステップ 2 **$ echo "@cert-authority * $(cat /etc/ssh/ca.pub)" > /etc/ssh/known_hosts**

例：

```
$ echo "@cert-authority * $(cat /etc/ssh/ca.pub)" > /etc/ssh/known_hosts
```

known_hosts に CA 公開キーを入力します

`@cert-authority` ディレクティブを使用して、 `known_hosts` ファイルに CA 公開キーを追加します。この手順によって、SSH クライアントがこの CA によって署名されたホスト証明書を信頼するようになります。

ステップ 3 **$ cat ~/.ssh/key_rsa.pub**

例：

```
$ cat ~/.ssh/key_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQE... user@host
```

公開キーの表示

ユーザー公開キーファイルの内容を表示します。このキーは証明書ベースの認証に必要であり、秘密キーおよび証明書に対応している必要があります。

（注）
キーの名前や場所が異なる場合は、それに応じてパスを調整します。

ステップ 4 **$ ssh -i <path_to_private_key> <user>@<hiba_server_ip>**

例：

```
$ ssh -i <path_to_private_key> <user>@<hiba_server_ip>
```

HIBA 対応 SSH サーバーへの接続

自分の秘密キー（および必要な場合は秘密キーとそれに一致する証明書）を使用して、SSH サーバーに接続します。

（注）
`-i` オプションは、ユーザーの秘密キー（アイデンティティ ファイル）を指定します。

正しく設定されている場合、 SSH 接続は HIBA 証明書ベースの認証を使用して確立され、CA 公開キーに対するホストの検証が成功します。公開キーがサーバーの `authorized_keys` に存在する場合、パスワードレス ログインが可能になります。

HIBA 構成の確認

手順

ステップ 1 **show crypto ca certificates type ssh**

例：

```
switch(config)# show crypto ca certificates type ssh
            trustpoint: openssh-ca
            CA Public Key:
            ecdsa-sha2-nistp384
AAAAE2VjZHNhLXNoYTItbmlzdHAzODQAAAAIbmlzdHAzODQAAABhBP[illegible]MQXA=
```

```
/home/admin/.hiba-ca CA
           Finger Print:
           384 SHA256:ZcJws/mPrts6twB29OoZU/c3AMAL0x3mUp00YxwSRmk /home/admin/.hiba-ca
CA (ECDSA)

           Host Certificate:
           Type: ecdsa-sha2-nistp384-cert-v01@openssh.com host certificate
           Public key: ECDSA-CERT SHA256:bZkNWnvyxUK1DHRwqayWivobGUwA25GRGkUMNEd/Ujw
           Signing CA: ECDSA SHA256:ZcJws/mPrts6twB29OoZU/c3AMAL0x3mUp00YxwSRmk (using
ecdsa-sha2-nistp384)
           Key ID: "cisco_nexus_9000"
           Serial: 1
           Valid: from 2025-06-05T04:34:00 to 2025-08-28T04:35:39
           Principals:
           cisco_nexus_9000
           Critical Options: (none)
           Extensions:
           identity@hibassh.dev

           HIBA Info:
           certificate 'cisco_nexus_9000' (1 principal) contains 1 HIBA grant
           principal: 'cisco_nexus_9000'
           identity@hibassh.dev (v2):
           [0] domain = 'google.com'
```

SSH証明書を表示し、ホスト証明書が登録済みで、正しいトラストポイント（openssh-ca）に関連付けられていることを確認します。

[想定される出力（Expected Output）]：出力には、HIBA 付与を示す「HIBA Info」セクションを含む、 SSH ホスト証明書の詳細が表示される必要があります。

ホスト証明書と HIBA 情報が正しく表示されれば、証明書の登録は成功です。

ステップ 2 show crypto ca trustpoints type ssh

例：

```
switch(config)# show crypto ca trustpoints type ssh
           trustpoint: openssh-ca
```

SSH トラストポイントを表示し、HIBA CA トラストポイント（openssh-ca）が存在することを確認します。

[想定される出力（Expected Output）]：出力には、タイプ sshのトラストポイント名が一覧表示されます。

HIBA CA トラストポイントが出力に表示される場合、トラストポイントは正常に構成されています。

ステップ 3 ssh -i path_to_private_key> <user>@<switch_ip

例：

```
ssh -i /home/admin/.hiba-ca/users/google-user admin@10.126.67.44
```

CA によって署名された HIBA 対応証明書を持つユーザーを使用して、スイッチに SSH で接続します。

注：**-i** オプションは、ユーザーの 秘密キー （アイデンティティ ファイル）へのパスを指定します。HIBA 拡張は、この秘密キーとペアになる証明書に含める必要があり、 CA 公開キーはスイッチによって信頼されている必要があります。秘密キーファイルが安全に保たれていることを確認してください。

パスワードの入力を求めずに、SSH接続が正常に確立されます（パスワード認証が無効になっている場合）。

Telnet サーバ

Telnet プロトコルは、ホストとの TCP/IP 接続を確立します。Telnet を使用すると、あるサイトのユーザが別のサイトのログイン サーバと TCP 接続を確立し、キーストロークをデバイス間でやり取りできます。Telnet は、リモート デバイス アドレスとして IP アドレスまたはドメイン名のいずれかを受け入れます。

デフォルトでは、Telnet サーバが Cisco NX-OS デバイス上でディセーブルになっています。

SSH および Telnet の前提条件

レイヤ 3 インターフェイス上で IP、mgmt 0 インターフェイス上でアウトバンド、またはイーサネット インターフェイス上でインバンドを設定していることを確認します。

SSH と Telnet のガイドラインと制約事項

SSH および Telnet に関する注意事項と制約事項は次のとおりです。

- Cisco NX-OS ソフトウェアは、SSH バージョン 2（SSHv2）だけをサポートしています。
- Cisco NX-OS リリース 10.4(3)F 以降、Cisco Nexus 9000 シリーズ スイッチは、TACACS+ サーバを介し、X.509 証明書を使用した SSH 認証をサポートしています。この機能は、RADIUS ではサポートされていません。
- **no feature ssh feature** コマンドを使用すると、ポート 22 はディセーブルになりません。ポート 22 は常にオープンで、すべての着信外部接続を拒否する拒否ルールがプッシュされます。
- Poodle の脆弱性により、SSLv3 はサポートされなくなりました。
- IPSG は、次のものではサポートされません。
  - Cisco Nexus 9372PX、9372TX、および 9332PQ スイッチの最後の 6 個の 40 Gb 物理ポート
  - Cisco Nexus 9396PX、9396TX、および 93128TX スイッチのすべての 40G 物理ポート

- X.509 証明書を使用する SSH 認証用にデバイスを設定できます。認証に失敗した場合は、パスワードの入力が求められます。
- SFTP サーバー機能は、通常の SFTP の **chown** および **chgrp** コマンドをサポートしません。
- SFTP サーバーが有効になっている場合は、admin ユーザだけが SFTP を使用してデバイスにアクセスできます。
- SSH パスワードレス ファイル コピーを目的として AAA プロトコル（RADIUS や TACACS+ など）を介してリモート認証されたユーザ アカウントにインポートされた SSH 公開キーと秘密キーは、同じ名前のローカル ユーザ アカウントでない限り、Nexus デバイスがリロードされると保持されません。リモート ユーザ アカウントは、SSH キーがインポートされる前にデバイスで設定されます。
- SSH のタイムアウト時間は、tac-pac の生成時間よりも長くする必要があります。そうでないと、VSH ログに %VSHD-2-VSHD_SYSLOG_EOL_ERR エラーが記録されることがあります。理想的には、tac-pac または showtech を収集する前に 0（無限）に設定します。
- Cisco NX-OS リリース 10.4(3)F 以降、**show running-config all** コマンドは次のコマンドの詳細を表示しません：
  - no feature telnet
  - no feature nxdb
  - no feature scp-server
  - no feature sftp-server

（注） Cisco IOS の CLI に慣れている場合、この機能の Cisco NX-OS コマンドは従来の Cisco IOS コマンドと異なる点があるため注意が必要です。

- Cisco NX-OS Release 10.2(2)F 以降、新しい非同期化 CLI が導入され、SNMP とセキュリティ コンポーネントの間のユーザー同期を無効にするオプションを提供します。詳細については、システム管理構成ガイドの *SNMP* の構成の章を参照してください。

  リリース 7.0(3)I7(1) から現在のリリースまでのさまざまな機能をサポートする Cisco Nexus 9000 スイッチの詳細については、Nexus スイッチ プラットフォーム サポート マトリックスを参照してください。
- 非同期 CLI が有効になっている場合、リモート ユーザーは SNMP データベースに同期されません。
- DCNM（リリース 12.0.1.a 以降 Nexus Dashboard Fabric Controller とも呼ばれる）を使用したセキュリティユーザーには、非同期 CLI が有効でないとき、対応する SNMPv3 プロファイルが存在しません。同期が無効になっている場合、セキュリティ コンポーネントで作成されたユーザーはスイッチにログインできますが、コントローラはスイッチを検出しません。コントローラは、セキュリティユーザー用に作成された SNMP 構成を使用してスイッチを検出するためです。さらに、SNMP は、userDB の非同期状態のため、作成されたセ

キュリティユーザーを認識しないので、スイッチを検出できません。したがって、コントローラによってスイッチが検出されるようにするには、SNMPユーザーを明示的に作成する必要があります。DCNM 機能とともに非同期 CLI を使用することはお勧めしません。詳細については、*Cisco Nexus 9000 NX-OS* セキュリティ構成ガイドを参照してください。

- Cisco NX-OS リリース 10.6(1)F 以降、次の DSA アルゴリズムとすべての DSA 関連の SSH CLI コマンドは廃止されています：
  - **show ssh key dsa**
  - **[no] ssh key dsa**
  - **[no] username username keypair generate {dsa [force]}**
  - **username username keypair export {bootflash:filename | volatile:filename} {dsa} [force]**
  - **username username keypair import {bootflash:filename | volatile:filename} {dsa} [force]**
  - **username user-id ssh-cert-dn dn-name {dsa}**
  - **[no] ssh cipher-mode weak**
  - **ssh ciphers aes256-gcm** ：このコマンドを実行すると、「未定義のアルゴリズム名（Undefined algorithm name）」という警告が表示されます。

（注） 廃止されたDSAおよび暗号モードCLIの場合、CLI構成の置換、デュアルステージ コミット、およびNetconf操作中に、廃止の警告は表示されません。セキュリティを強化するため、SSH認証と管理に RSA キーまたは ECDSA キーを生成して使用します。

- Cisco NX-OS リリース 10.6(2)F 以降、 Cisco Nexus 9000 シリーズ スイッチは、既存の netstack パスに加えて、 Linux カーネル ネットワークスタック（kstack）を介した SSH 接続の受け入れをサポートしています。

SSH および Telnet のデフォルト設定

次の表に、SSH および Telnet パラメータのデフォルト設定を示します。

Table 1: デフォルトの ***SSH*** および ***Telnet*** パラメータ

パラメータ	デフォルト
SSH サーバ	イネーブル
SSH サーバ キー	1024 ビットで生成された RSA キー
RSA キー生成ビット数	1024

パラメータ	デフォルト
Telnet サーバ	無効化
Telnet ポート番号	23
SSH ログインの最大試行回数	3
SCP サーバ	ディセーブル
SFTP サーバ	無効化

SSH の設定

ここでは、SSH の設定方法について説明します。

SSH サーバ キーの生成

セキュリティ要件に基づいて SSH サーバ キーを生成できます。デフォルトの SSH サーバ キーは、1024 ビットで生成される RSA キーです。

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	SSH を無効にします。

ステップ 3	**ssh key export** *export-host-keypath* {**rsa** | **ecdsa**} [**force**]
	Example:
	`switch(config)# ssh key rsa export bootflash:host_key rsa`
	`Enter Passphrase:`
		SSH サーバー キーをエクスポートします。
		SSHサーバー キーを既存のファイル パスにエクスポートする場合は、 **force** キーワードを使用します。

ステップ 4	**ssh rekey max-data** *max-data* **max-time** *max-time*i **Example:** `switch(config)# ssh rekey max-data 1K max-time 1M`	キー再生成パラメータを設定します。

	Command or Action	Purpose
ステップ 5	**feature ssh** **Example:** `switch(config)# feature ssh`	SSH を有効にします。

ステップ 6	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		グローバル コンフィギュレーション モードを終了します。

ステップ 7	(Optional) **show ssh key** [**dsa** | **rsa** | **ecdsa**] [**md5**] **Example:** `switch# show ssh key`	SSH サーバ キーを表示します。 このコマンドは、デフォルトでSHA256形式でフィンガープリントを表示します。SHA256 は、以前のデフォルトのMD5形式よりも安全です。ただし、フィンガープリントを MD5 形式で表示する必要がある場合の下位互換性のために、**md5** オプションが追加されています。 **Note** Cisco NX-OSリリース 10.6(1)F 以降、dsa キーワードは非表示になっています。フル コマンドを入力すると、NX- OS は廃止の警告を出しますが、それを受け入れます。
ステップ 8	**show run security all**	
ステップ 9	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

ユーザ アカウント用 SSH 公開キーの指定

SSH公開キーを設定すると、パスワードを要求されることなく、SSHクライアントを使用してログインできます。SSH 公開キーは、次のいずれかの形式で指定できます。

- OpenSSH 形式
- Internet Engineering Task Force（IETF）SECSH 形式

IETF SECSH 形式による SSH 公開キーの指定

ユーザ アカウント用に IETF SECSH 形式で SSH 公開キーを指定できます。

Before you begin

IETF SCHSH 形式の SSH 公開キーを作成します。

Procedure

	Command or Action	Purpose
ステップ 1	**copy** *server-file* **bootflash:***filename* **Example:** `switch# copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub`	サーバから IETF SECSH 形式の SSH キーを含むファイルをダウンロードします。サーバは FTP、Secure Copy（SCP）、Secure FTP（SFTP）、または TFTP のいずれかを使用できます。
ステップ 2	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	グローバル 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	**username** *username* **sshkey file bootflash:***filename* **Example:** `switch(config)# username User1 sshkey file bootflash:secsh_file.pub`	IETF SECSH 形式の SSH 公開キーを設定します。
ステップ 4	**exit** **Example:** `switch(config)# exit switch#`	グローバル コンフィギュレーション モードを終了します。
ステップ 5	(Optional) **show user-account** **Example:** `switch# show user-account`	ユーザアカウントの設定を表示します。
ステップ 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

OpenSSH 形式の SSH 公開キーの指定

ユーザ アカウントに OpenSSH 形式の SSH 公開キーを指定できます。

Before you begin

OpenSSH 形式の SSH 公開キーを作成します。

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**username** *username* **sshkey** *ssh-key*
	Example:
	`switch(config)# username User1 sshkey ssh-rsa`
	`AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50rv7gsEPj`
	`hOBYmsi6PAVKui1nIf/DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQW3g9igG30c6k6+`
	`XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH3UD/vKyziEh5S4Tplx8=`
		OpenSSH 形式の SSH 公開キーを設定します。

ステップ 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		グローバル コンフィギュレーション モードを終了します。

ステップ 4	(Optional) **show user-account** **Example:** `switch# show user-account`	ユーザアカウントの設定を表示します。
ステップ 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SSH ログイン試行の最大回数の設定

SSH ログイン試行の最大回数を設定できます。許可される試行の最大回数を超えると、セッションが切断されます。

Note ログイン試行の合計回数には、公開キー認証、証明書ベースの認証、およびパスワードベースの認証を使用した試行が含まれます。イネーブルにされている場合は、公開キー認証が優先されます。証明書ベースとパスワードベースの認証だけがイネーブルにされている場合は、証明書ベースの認証が優先されます。これらすべての方法で、ログイン試行の設定された数を超えると、認証失敗回数を超過したことを示すメッセージが表示されます。

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**ssh login-attempts** *number* **Example:** `switch(config)# ssh login-attempts 5`	ユーザが SSH セッションへのログインを試行できる最大回数を設定します。ログイン試行のデフォルトの最大回数は3です。値の範囲は 1 ～ 10 です。 **Note** このコマンドの **no** 形式を使用すると、以前のログイン試行の値が削除され、ログイン試行の最大回数がデフォルト値の 3 に設定されます。
ステップ 3	(Optional) **show running-config security all** **Example:** `switch(config)# show running-config security all`	SSH ログイン試行の設定された最大回数を表示します。
ステップ 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

SSH セッションの開始

Cisco NX-OS デバイスから IPv4 または IPv6 を使用して SSH セッションを開始し、リモート デバイスと接続します。

Before you begin

リモート デバイスのホスト名を取得し、必要なら、リモート デバイスのユーザ名も取得します。

リモート デバイスの SSH サーバを有効にします。

Procedure

	Command or Action	Purpose
ステップ 1	**ssh** [*username@*]{*ipv4-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `switch# ssh 10.10.1.1`	IPv4 を使用してリモート デバイスとの SSH IPv4 セッションを作成します。デフォルトの VRF はデフォルト VRF です。
ステップ 2	**ssh6** [*username@*]{*ipv6-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `switch# ssh6 HostA`	IPv6 を使用してリモート デバイスとの SSH IPv6 セッションを作成します。

ブート モードからの SSH セッションの開始

SSH セッションは、リモート デバイスに接続する Cisco NX-OS デバイスのブート モードから開始できます。

Before you begin

リモート デバイスのホスト名を取得し、必要なら、リモート デバイスのユーザ名も取得します。

リモート デバイスの SSH サーバを有効にします。

Procedure

	Command or Action	Purpose
ステップ 1	**ssh** [*username@*]*hostname* **Example:** `switch(boot)# ssh user1@10.10.1.1`	リモート デバイスへの SSH セッションを、Cisco NX-OS デバイスのブート モードから作成します。デフォルト VRF が常に使用されます。
ステップ 2	**exit** **Example:** `switch(boot)# exit`	ブート モードを終了します。
ステップ 3	**copy scp:**//[*username@*]*hostname*/*filepath directory* **Example:** `switch# copy scp://user1@10.10.1.1/users abc`	セキュア コピー プロトコル（SCP）を使用して、ファイルを Cisco NX-OS デバイスからリモート デバイスへコピーします。デフォルト VRF が常に使用されます。

SSH のパスワードが不要なファイル コピーの設定

Cisco NX-OS デバイスから Secure Copy（SCP）サーバまたは Secure FTP（SFTP）サーバに、パスワードなしでファイルをコピーすることができます。これを行うには、SSH による認証用の公開キーと秘密キーで構成される RSA または DSA のアイデンティティを作成する必要があります。

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	[**no**] **username** *username* **keypair generate** {**rsa** [*bits* [**force**]] | **dsa** [**force**]} **Example:** `switch(config)# username user1 keypair generate rsa 2048 force`	SSH の公開キーと秘密キーを生成し、指定したユーザの Cisco NX-OS デバイスのホーム ディレクトリ（$HOME/.ssh）に格納します。Cisco NX-OS デバイスでは、これらのキーを使用してリモート マシンの SSH サーバと通信します。 *bits* 引数には、キーの生成に使用するビット数を指定します。有効な範囲は 768 ～ 4096 です。デフォルト値は 1024 です。 既存のキーを置き換える場合は、**force** キーワードを使用します。**force** キーワードを省略した場合、SSH キーがすでに存在していれば、SSH キーは生成されません。 **Note** Cisco NX-OS リリース 10.6(1)F 以降、dsa キーワードは非表示になっています。フル コマンドを入力すると、NX- OS は廃止の警告を出しますが、それを受け入れます。
ステップ 3	(Optional) **show username** *username* **keypair** **Example:** `switch(config)# show username user1 keypair`	指定したユーザの公開キーを表示します。 **Note** セキュリティ上の理由から、このコマンドで秘密キーは表示されません。

	Command or Action	Purpose
ステップ 4	Required: **username** *username* **keypair export** {**bootflash:***filename* | **volatile:***filename*} {**rsa** | **dsa**} [**force**] **Example:** `switch(config)# username user1 keypair export bootflash:key_rsa rsa`	Cisco NX-OS デバイスのホーム ディレクトリから、指定したブートフラッシュ ディレクトリまたは一時ディレクトリに、公開キーと秘密キーをエクスポートします。 既存のキーを置き換える場合は、**force** キーワードを使用します。**force** キーワードを省略した場合、SSH キーがすでに存在していれば、SSH キーはエクスポートされません。 生成したキー ペアをエクスポートするとき、秘密キーを暗号化するパスフレーズを入力するように求められます。秘密キーは、指定したファイルとしてエクスポートされ、公開キーは、同じファイル名に .pub 拡張子を付けてエクスポートされます。これで、このキー ペアを任意の Cisco NX-OS デバイスにコピーし、SCP または SFTP を使用してサーバのホーム ディレクトリに公開キー ファイル（*.pub）をコピーできるようになります。 **Note** セキュリティ上の理由から、このコマンドはグローバル コンフィギュレーション モードでしか実行できません。 **Note** Cisco NX-OS リリース 10.6(1)F 以降、dsa キーワードは非表示になっています。フル コマンドを入力すると、NX- OS は廃止の警告を出しますが、それを受け入れます。
ステップ 5	Required: **username** *username* **keypair import** {**bootflash:***filename* | **volatile:***filename*} {**rsa** | **dsa**} [**force**] **Example:** `switch(config)# username user1 keypair import bootflash:key_rsa rsa`	指定したブートフラッシュ ディレクトリまたは一時ディレクトリから、Cisco NX-OS デバイスのホーム ディレクトリに、エクスポートした公開キーと秘密キーをインポートします。 **Note** Cisco NX-OS リリース 10.6(1)F 以降、dsa キーワードは非表示になっています。

	Command or Action	Purpose
		フル コマンドを入力すると、NX- OS は廃止の警告を出しますが、それを受け入れます。 既存のキーを置き換える場合は、**force** キーワードを使用します。**force** キーワードを省略した場合、SSH キーがすでに存在していれば、SSH キーはインポートされません。 生成したキー ペアをインポートするとき、秘密キーを復号化するパスフレーズを入力するように求められます。秘密キーは指定したファイルとしてインポートされ、公開キーは同じファイル名に .pub 拡張子を付けてインポートされます。 **Note** セキュリティ上の理由から、このコマンドはグローバル コンフィギュレーション モードでしか実行できません。 **Note** パスワードなしでサーバにアクセスできるのは、サーバでキーが設定されているユーザのみです。

What to do next

SCP サーバまたは SFTP サーバで、次のコマンドを使用して、*.pub ファイル（たとえば、key_rsa.pub）に格納された公開キーを authorized_keys ファイルに追加します。

$ cat key_rsa.pub >> $HOME/.ssh/ authorized_keys

これで、標準の SSH コマンドおよび SCP コマンドを使用してパスワードを指定しなくても、Cisco NX-OS デバイスからサーバにファイルをコピーできます。

SCP サーバと SFTP サーバの設定

リモートデバイスとの間でファイルをコピーできるように、Cisco NX-OS デバイスで SCP サーバまたは SFTP サーバを設定できます。SCP サーバまたは SFTP サーバをイネーブルにした後、Cisco NX-OS デバイスとの間でファイルをコピーするために、リモート デバイスで SCP または SFTP コマンドを実行できます。

Note arcfour および blowfish cipher オプションは SCP サーバではサポートされません。

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

ステップ 2	[**no**] **feature scp-server** **Example:** `switch(config)# feature scp-server`	Cisco NX-OS デバイス上で SCP サーバをイネーブルまたはディセーブルにします。
ステップ 3	Required: [**no**] **feature sftp-server** **Example:** `switch(config)# feature sftp-server`	Cisco NX-OS デバイス上で SFTP サーバをイネーブルまたはディセーブルにします。

ステップ 4	Required: **exit**
	Example:
	`switch(config)# exit`
	`switch#`
		グローバル コンフィギュレーション モードを終了します。

ステップ 5	(Optional) **show running-config security** **Example:** `switch# show running-config security`	SCP サーバと SFTP サーバの設定ステータスを表示します。
ステップ 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

X.509v3 証明書ベースの SSH 認証の設定

X.509v3 証明書を使用する SSH 認証を設定できます。

始める前に

リモート デバイスの SSH サーバをイネーブルにします。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します

ステップ 2	**username** *user-id* [**password** [**0** | **5**] *password*] 例： `switch(config)# username jsmith password 4Ty18Rnt`	ユーザ アカウントを設定します。*user-id* 引数は、大文字と小文字が区別される英数字で、最大 28 文字です。これはローカルおよびリモートユーザーの両方に当てはまります。指定できる文字は、A ～ Z の英大文字、a ～ z の英小文字、0 ～ 9 の数字、ハイフン（-）、ピリオド（.）、アンダースコア（_）、プラス符号（+）、および等号（=）です。アット マーク（@）はリモートユーザ名では使用できますが、ローカル ユーザ名では使用できません。 ユーザ名の先頭は英数字で始まる必要があります。 デフォルトパスワードは定義されていません。オプションの **0** は、パスワードがクリア テキストであり、**5** はパスワードが暗号化されていることを意味します。デフォルトは **0**（クリア テキスト）です。 （注） パスワードを指定しなかった場合、ユーザは Cisco NX-OS デバイスにログインできません。 （注） 暗号化パスワード オプションを使用してユーザアカウントを作成する場合、対応する SNMP ユーザは作成されません。 （注） 非同期 CLI が有効になっている場合、ユーザーアカウントを作成しても、対

	コマンドまたはアクション	目的
		応する SNMP ユーザーは作成されません。
ステップ 3	**username** *user-id* **ssh-cert-dn** *dn-name* {**dsa** | **rsa**} 例： `switch(config)# username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1, emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith" rsa`	既存のユーザアカウント認証に使用する SSH X.509 証明書の識別名と DSA アルゴリズムを指定します。識別名は最大 512 文字で、例に示す形式に従う必要があります。電子メールアドレスと状態がそれぞれ emailAddress と ST に設定されていることを確認します。

ステップ 4	[**no**] **crypto ca trustpoint** *trustpoint*
	例：
	`switch(config)# crypto ca trustpoint winca`
	`switch(config-trustpoint)#`
		トラストポイントを設定します。
		（注）
		このコマンドの **no** 形式を使用してトラストポイントを削除する前に、まず **delete crl** および **delete ca-certificate** コマンドを使用して、CRL および CA 証明書を削除する必要があります。

ステップ 5	**crypto ca authenticate** *trustpoint* 例： `switch(config-trustpoint)# crypto ca authenticate winca`	トラストポイントの CA 証明書を設定します。 （注） CA 証明書を削除するには、トラストポイントコンフィギュレーションモードで **delete ca-certificate** コマンドを入力します。
ステップ 6	（任意） **crypto ca crl request** *trustpoint* **bootflash:***static-crl***.crl** 例： `switch(config-trustpoint)# crypto ca crl request winca bootflash:crllist.crl`	この項はオプションですが、強く推奨されます。トラストポイントの証明書失効リスト（CRL）を設定します。CRL ファイルは、トラストポイントによって失効した証明書のリストのスナップショットです。このスタティック CRL リストは、認証局（CA）からデバイスに手動でコピーされます。 （注） スタティック CRL は、サポートされている唯一の失効チェック方式です。 （注） CRL を削除するには、**delete crl** コマンドを入力します。

	コマンドまたはアクション	目的
ステップ 7	（任意） **show crypto ca certificates** 例： `switch(config-trustpoint)# show crypto ca certificates`	設定されている証明書またはチェーンと、関連付けられているトラストポイントを表示します。
ステップ 8	（任意） **show crypto ca crl** *trustpoint* 例： `switch(config-trustpoint)# show crypto ca crl winca`	指定したトラストポイントの CRL リストの内容を表示します。
ステップ 9	（任意） **show user-account** 例： `switch(config-trustpoint)# show user-account`	設定されたユーザアカウントの詳細を表示します。
ステップ 10	（任意） **show users** 例： `switch(config-trustpoint)# show users`	デバイスにログオンしているユーザが表示されます。
ステップ 11	（任意） **copy running-config startup-config** 例： `switch(config-trustpoint)# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップコンフィギュレーションにコピーします。

TACACS サーバでの SSH 証明書認証の構成

Cisco NX-OS リリース 10.4(3)F 以降、Cisco Nexus 9000 シリーズ スイッチは、TACACS+ サーバを介し、X.509 証明書を使用した SSH 認証をサポートしています。この機能は、RADIUS ではサポートされていません。この機能は、 **aaa authorization ssh-certificate default group** *tac-group-name* コマンドを使用して有効にできます。設定の詳細については、TACACS サーバでの AAA SSH 証明書認証の構成 を参照してください。

SSH 暗号化アルゴリズムのカスタマイズ

Cisco Nexus 9000 スイッチは、デフォルトで強力なアルゴリズムをサポートします。Cisco Product Security Baseline（PSB）で定義されている強力なアルゴリズムのみを有効にするデフォルトモードのままにするか、サポートされているすべてのアルゴリズムを許可するかを選択できます。これらのアルゴリズムは、着信サーバー接続に適用できることに注意してください。SSH キー交換アルゴリズム、メッセージ認証コード（MAC）、キー タイプ、および暗号のサポートを設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	（任意） **ssh kexalgos** [**all** | **key-exchangealgorithm-name**] 例： `switch(config)# ssh kexalgos ecdhsha2-nistp384`	接続ごとのキーの生成に使用されるキー交換方式である、サポートされているすべての KexAlgorithm を有効にするには、**all** キーワードを使用します。 サポートされる KexAlgorithm は次のとおりです。 • curve25519-sha256 • curve25519-sha256@libssh.org • diffie-hellman-group14-sha1 • diffie-hellman-group14-sha256 • diffie-hellman-group16-sha512 • ecdh-sha2-nistp256 • ecdh-sha2-nistp521 サポートされない KexAlgorithm は次のとおりです。 • diffie-hellman-group1-sha1 • diffie-hellman-group-exchange-sha256 **ecdh-sha2-nistp384** KexAlgorithm のみを有効にするには、 **ecdh-sha2-nistp384** キーワードを使用します。 （注） Cisco NX-OS リリース 10.4(2) 以降では、サポートされている任意の KexAlgorithm を設定できます。このリリースから、**ecdh-sha2-nistp384** キーワードは廃止されました。
ステップ 3	（任意） **ssh macs [all | mac-name]** 例：	トラフィック変更の検出に使用されるメッセージ認証コードである、サポート

	コマンドまたはアクション	目的
	`switch(config)# ssh macs hmacsha2-256-etm@openssh.com`	されているすべての MAC を有効にします。 サポートされる MAC は次のとおりです。 • hmac-sha1 • hmac-sha2-256 • hmac-sha2-512 • hmac-sha2-256-etm@openssh.com • hmac-sha2-512-etm@openssh.com • hmac-sha1-etm@openssh.com （注） Cisco NX-OS リリース 10.4(2)F 以降では、サポートされている任意の MAC を設定できます。
ステップ 4	（任意） **ssh ciphers** [**all** | **cipher-name**] 例： `switch(config)# ssh ciphers aes192-ctr`	サポートされているすべての暗号を有効にして接続を暗号化するには、 **all** キーワードを使用します。 サポート対象の暗号方式： • aes128-cbc • aes192-cbc • aes256-cbc • aes128-ctr • aes192-ctr • aes256-ctr • aes128-gcm@openssh.com • chacha20-poly1305@openssh.com （注） Cisco NX-OS リリース 10.4(2)F 以降では、サポートされている任意の暗号を設定できます。このリリースから、 **aes256-gcm** キーワードは廃止されました。 （注）

	コマンドまたはアクション	目的
		Cisco NX-OSリリース 10.6(1) 以降、`ssh ciphers aes256-gcm` コマンドは廃止され、サポートされなくなりました。このコマンドが設定に存在する場合、設定の置換（CR）とISSUは失敗します。アップグレードする前に設定からコマンドを削除してください。

ステップ 5	（任意） **ssh keytypes [all | keytype-string]**
	例：
	`switch(config)# ssh keytypes ecdsa-sha2-nistp256`
		サーバーがクライアントに対して自身を認証するために使用できる公開キー アルゴリズムである、サポートされているすべての PubkeyAcceptedKeyType を有効にします。
		サポートされるキー タイプは次のとおりです。
		• ecdsa-sha2-nistp256
		• ecdsa-sha2-nistp384
		• ecdsa-sha2-nistp521
		• ecdsa-sha2-nistp256-cert-v01@openssh.com
		• ecdsa-sha2-nistp384-cert-v01@openssh.com
		• ecdsa-sha2-nistp521-cert-v01@openssh.com
		• ssh-dss
		• rsa-sha2-256
		• ssh-rsa-cert-v01@openssh.com
		• ssh-rsa
		（注）
		Cisco NX-OS リリース 10.4(2)F 以降では、サポートされている任意のキータイプを設定できます。
		（注）
		rsa、dsa、および ecdsa キータイプを有効にするには、対応する SSH ホスト キーを生成する必要があります。
		設定例：
		`switch(config)# ssh key rsa 2048`
		`switch(config)# ssh key dsa`

	コマンドまたはアクション	目的
		switch(config)# ssh key ecdsa 256

例

show ssh [ciphers | macs | keytypes | kexalogs | version] コマンドを使用して、サポートされているアルゴリズムを確認できます。

```
show ssh ciphers
Cipher                                          Status    FIPS
-------------------------------------------------------------
aes128-ctr                                   permitted     yes
aes192-ctr                                      denied     yes
aes256-ctr                                   permitted     yes
aes128-cbc                                      denied     yes
aes192-cbc                                      denied     yes
aes256-cbc                                      denied     yes
aes256-gcm@openssh.com                    permitted     yes
aes128-gcm@openssh.com                       permitted     yes
chacha20-poly1305@openssh.com                permitted      no
```

```
show ssh macMAC                                                 Status    FIPS
-------------------------------------------------------------
hmac-sha2-256-etm@openssh.com                permitted      no
hmac-sha2-512-etm@openssh.com                permitted      no
hmac-sha1-etm@openssh.com                    permitted      no
hmac-sha2-256                                permitted     yes
hmac-sha2-512                                permitted     yes
hmac-sha1                                    permitted     yes
hmac-sha1-96                              unsupported      no
hmac-md5                                  unsupported      no
hmac-md5-96                               unsupported      no
umac-64@openssh.com                       unsupported      no
umac-128@openssh.com                      unsupported      no
hmac-sha1-96-etm@openssh.com              unsupported      no
hmac-md5-etm@openssh.com                  unsupported      no
umac-64-etm@openssh.com                   unsupported      no
umac-128-etm@openssh.com                  unsupported      no
```

```
show ssh keytypes Keytype                                          Status    FIPS
-------------------------------------------------------------
ecdsa-sha2-nistp256-cert-v01@openssh.com   permitted     no <<Currently not suppported>>
ecdsa-sha2-nistp384-cert-v01@openssh.com   permitted     no <<Currently not suppported>>
ecdsa-sha2-nistp521-cert-v01@openssh.com   permitted     no <<Currently not suppported>>
ssh-rsa-cert-v01@openssh.com                permitted      no
ecdsa-sha2-nistp256                         permitted     yes
ecdsa-sha2-nistp384                         permitted     yes
ecdsa-sha2-nistp521                         permitted      no
rsa-sha2-256                                permitted      no
ssh-rsa                                     permitted     yes
ssh-dss                                        denied      no
ssh-ed25519                               unsupported      no
ssh-ed25519-cert-v01@openssh.com          unsupported      no
ssh-dss-cert-v01@openssh.com              unsupported      no
```

```
show ssh kexalgos
KexAlgorithm                                        Status    FIPS
-------------------------------------------------------------------
curve25519-sha256                                 permitted      no
curve25519-sha256@libssh.org                      permitted      no
ecdh-sha2-nistp256                                permitted     yes
ecdh-sha2-nistp384                                permitted     yes
ecdh-sha2-nistp521                                permitted     yes
diffie-hellman-group16-sha512                     permitted     yes
diffie-hellman-group14-sha1                       permitted     yes
diffie-hellman-group14-sha256                     permitted      no


show ssh version
CiscoSSH 1.9.29, OpenSSH_8.3p1, CiscoSSL 1.1.1t.7.2.500
```

サポートされるアルゴリズム：FIPモードが有効の場合

FIP モードが有効な場合にサポートされるアルゴリズムのリストは次のとおりです。

表 *2:* サポートされるアルゴリズム：*FIP*モードが有効の場合

アルゴリズム	サポート対象	サポート対象外
ciphers	• aes128-ctr • aes256-ctr • aes256-gcm@openssh.com • aes128-gcm@openssh.com • chacha20-poly1305@openssh.com	• aes192-ctr • aes128-cbc • aes192-cbc • aes256-cbc
hmac	• hmac-sha2-256 • hmac-sha2-512 • hmac-sha1 • hmac-sha2-256-etm@openssh.com • hmac-sha2-512-etm@openssh.com • hmac-sha1-etm@openssh.com	-

アルゴリズム	サポート対象	サポート対象外
kexalgo	• ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521 • diffie-hellman-group16-sha512 • diffie-hellman-group14-sha1 • diffie-hellman-group14-sha256 • curve25519-sha256@libssh.org • curve25519-sha256	-
keytypes	• ecdsa-sha2-nistp256-cert-v01@openssh.com • ecdsa-sha2-nistp384-cert-v01@openssh.com • ecdsa-sha2-nistp521-cert-v01@openssh.com • ssh-rsa-cert-v01@openssh.com • rsa-sha2-256 • ssh-rsa • ecdsa-sha2-nistp256 • ecdsa-sha2-nistp384 • ecdsa-sha2-nistp521	• ssh-dss

デフォルトの SSH サーバ ポートの変更

Cisco NX-OS Cisco リリース 9.2(1) 以降では、SSHv2 のポート番号をデフォルトのポート番号 22 から変更できます。デフォルトの SSH ポートの変更時に使用される暗号化により、より強力なプライバシーとセッション整合性をサポートする接続が実現します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション 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	**no feature ssh** 例： `switch(config)# no feature ssh`	SSH を無効にします。

ステップ 3	**show sockets** *local-port-range*
	例：
	`switch(config)# show sockets local port range (15001 - 58000)`
	`switch(config)# local port range (58001 - 63535) and nat port range (63536 - 65535)`
	`switch# show sockets local-port-range`
	`Kstack local port range (15001 - 22002)`
	`Netstack local port range (22003 - 65535)`
		使用可能なポート範囲を表示します。

ステップ 4	**ssh port** *local-port* 例： `switch(config)# ssh port 58003`	ポートを設定します。 （注） 以前のリリースからリリース 9.3(1) 以降のリリースにアップグレードする場合は、ユーザ定義の SSH ポートを使用する機能が次の範囲内にあることを確認してください。 • リリース 9.3(1) およびリリース 9.3(2) の場合：Kstack ローカル ポートの範囲は 15001 ー 58000、netstack ローカル ポートの範囲は58001 ー 63535、nat ポートの範囲は 63536 ー 65535 • リリース 9.3(3) 以降：Kstack ローカルポートの範囲は 15001 ー 58000、netstack ローカル ポートの範囲は 58001 ー 60535 、nat ポートの範囲は 60536 ー 65535
ステップ 5	**feature ssh** 例： `switch(config)# feature ssh`	SSH を有効にします。

ステップ 6	**exit**
	例：
	`switch(config)# exit`
	`switch#`
		グローバル コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
ステップ 7	（任意） **show running-config security all** 例： `switch# ssh port 58003`	セキュリティの設定を表示します。
ステップ 8	（任意） **copy running-config startup-config** 例： `switch# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SSH ホストのクリア

サーバから SCP または SFTP を使用してファイルをダウンロードする場合、またはこのデバイスからリモート ホストに SSH セッションを開始する場合には、そのサーバと信頼できる SSH 関係が確立されます。ユーザ アカウントの、信頼できる SSH サーバのリストはクリアすることができます。

Procedure

	Command or Action	Purpose
ステップ 1	**clear ssh hosts** **Example:** `switch# clear ssh hosts`	SSH ホスト セッションおよび既知のホスト ファイルをクリアします。

SSH サーバのディセーブル化

Cisco NX-OS では、デフォルトで SSH サーバがイネーブルになっています。SSH サーバをディセーブルにすると、SSH でスイッチにアクセスすることを防止できます。

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	SSH を無効にします。

	Command or Action	Purpose

ステップ 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		グローバル コンフィギュレーション モードを終了します。

ステップ 4	(Optional) **show ssh server** **Example:** `switch# show ssh server`	SSH サーバの設定を表示します。
ステップ 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

SSH サーバ キーの削除

SSH サーバをディセーブルにした後、Cisco NX-OS デバイス上の SSH サーバ キーを削除できます。

Note SSH を再度イネーブルにするには、まず、SSH サーバ キーを生成する必要があります。

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	SSH を無効にします。
ステップ 3	**no ssh key**[**dsa** |**rsa** |**ecdsa**] **Example:** `switch(config)# no ssh key rsa`	SSH サーバ キーを削除します。 デフォルトでは、すべての SSH キーが削除されます。
ステップ 4	**exit** **Example:**	グローバル コンフィギュレーション モードを終了します。

	Command or Action	Purpose
	`switch(config)# exit` `switch#`	
ステップ 5	(Optional) **show ssh key** **Example:** `switch# show ssh key`	SSH サーバ キーの設定を表示します。
ステップ 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

Related Topics

SSH サーバ キーの生成 （13 ページ）

SSH セッションのクリア

Cisco NX-OS デバイスから SSH セッションをクリアできます。

Procedure

	Command or Action	Purpose
ステップ 1	**show users** **Example:** `switch# show users`	ユーザ セッション情報を表示します。
ステップ 2	**clear line** *vty-line* **Example:** `switch(config)# clear line pts/12`	ユーザSSHセッションをクリアします。

Telnet の設定

ここでは、Cisco NX-OS デバイスで Telnet を設定する手順を説明します。

Telnet サーバのイネーブル化

Cisco NX-OS デバイス上で Telnet サーバをイネーブルにできます。デフォルトでは、Telnet はディセーブルです。

Procedure

	Command or Action	Purpose

ステップ 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**feature telnet** **Example:** `switch(config)# feature telnet`	Telnet サーバをイネーブルにします。デフォルトではディセーブルになっています。

ステップ 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		グローバル コンフィギュレーション モードを終了します。

ステップ 4	(Optional) **show telnet server** **Example:** `switch# show telnet server`	Telnet サーバの設定を表示します。
ステップ 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	実行コンフィギュレーションを、スタートアップ コンフィギュレーションにコピーします。

リモート デバイスとの Telnet セッションの開始

Cisco NX-OS デバイスから SSH セッションを開始して、リモート デバイスと接続できます。IPv4 または IPv6 のいずれかを使用して Telnet セッションを開始できます。

Before you begin

リモート デバイスのホスト名または IP アドレスと、必要な場合はリモート デバイスのユーザ名を取得します。

Cisco NX-OS デバイス上で Telnet サーバを有効にします。

リモート デバイス上で Telnet サーバを有効にします。

Procedure

	Command or Action	Purpose
ステップ 1	**telnet** {*ipv4-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*] **Example:** `switch# telnet 10.10.1.1`	IPv4 を使用してリモート デバイスとの Telnet セッションを開始します。デフォルトのポート番号は 23 です。値の範囲は 1 ～ 65535 です。デフォルトの VRF はデフォルト VRF です。
ステップ 2	**telnet6** {*ipv6-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*] **Example:** `switch# telnet6 2001:0DB8::ABCD:1 vrf management`	IPv6 を使用してリモート デバイスとの Telnet セッションを開始します。デフォルトのポート番号は 23 です。値の範囲は 1 ～ 65535 です。デフォルトの VRF はデフォルト VRF です。

Related Topics

Telnet サーバのイネーブル化 （35 ページ）

Telnet セッションのクリア

Cisco NX-OS デバイスから Telnet セッションをクリアできます。

Before you begin

Cisco NX-OS デバイス上で Telnet サーバをイネーブルにします。

Procedure

	Command or Action	Purpose
ステップ 1	**show users** **Example:** `switch# show users`	ユーザ セッション情報を表示します。
ステップ 2	**clear line** *vty-line* **Example:** `switch(config)# clear line pts/12`	ユーザ Telnet セッションをクリアします。

SSH および Telnet の設定の確認

SSH および Telnet の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show ssh key [**dsa** | **rsa**] [**md5**]	SSH サーバ キーを表示します。 Cisco NX-OS リリース 7.0(3)I4(6) および 7.0(3)I6(1) 以降のリリースでは、このコマンドはデフォルトで SHA256 形式でフィンガープリントを表示します。SHA256 は、以前のデフォルトの MD5形式よりも安全です。ただし、フィンガープリントをMD5形式で表示する必要がある場合の下位互換性のために、**md5** オプションが追加されています。
show running-config security [**all**]	実行コンフィギュレーション内の SSH とユーザ アカウントの設定を表示します。**all** キーワードを指定すると、SSH およびユーザ アカウントのデフォルト値が表示されます。
show ssh server	SSH サーバの設定を表示します。
show telnet server	Telnet サーバの設定を表示します。
show username *username* **keypair**	指定したユーザの公開キーを表示します。
show user-account	設定されたユーザ アカウントの詳細を表示します。
show users	デバイスにログオンしているユーザが表示されます。
show crypto ca certificates	X.509v3証明書ベースのSSH認証に設定されたCA証明書および関連するトラストポイントを表示します。
show crypto ca crl *trustpoint*	指定したトラストポイントのCRL リストの内容を表示します。

SSH の設定例

次の例は、OpenSSH キーを使用して SSH を設定する方法を示しています。

Procedure

ステップ 1 SSH サーバをディセーブルにします。

Example:

```
switch# configure terminal
switch(config)# no feature ssh
```

ステップ 2 SSH サーバ キーを生成します。

Example:

```
switch(config)# ssh key rsa
generating rsa key(1024 bits)......
generated rsa key
```

ステップ 3 SSH サーバをイネーブルにします。

Example:

```
switch(config)# feature ssh
```

ステップ 4 SSH サーバ キーを表示します。

Example:

```
switch(config)# show ssh key
could not retrieve dsa key information
*************************************
rsa Keys generated:Tue Mar 14 13:13:47 2017

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDh4+DZboQJbJt10nJhgKBYL5lOlhsFM2oZRi9+JqEU
GA44I9ej+E5NIRZ1x8ohIt6Vx9Et5csO7Pw72rjUwR3UPmuAm79k7I/SyLGEP3WUL7sqbLvNF5GqKXph
oqMT075WUdbGWphorA2g0tTObRrFIQBJVQ0SSBh3oEaaALqYUQ==

bitcount:1024
fingerprint:
SHA256:V6KAeLAiKRRUPBZm1Yq3rl6JW7Eo7vhLi6CXYxnD/+Y
*************************************
*************************************


switch(config)# show ssh key
rsa Keys generated:Sat Sep 29 00:10:39 2013

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAvWhEBsF55oaPHNDBnpXOTw6+/OdHoLJZKr
+MZm99n2U0ChzZG4svRWmHuJY4PeDWl0e5yE3g3EO3pjDDmt923siNiv5aSga60K36lr39
HmXL6VgpRVn1XQFiBwn4na+H1d3Q0hDt+uWEA0tka2uOtXlDhliEmn4HVXOjGhFhoNE=

bitcount:1024
fingerprint:
51:6d:de:1c:c3:29:50:88:df:cc:95:f0:15:5d:9a:df
*************************************
could not retrieve dsa key information
*************************************
```

ステップ 5 OpenSSH 形式の SSH 公開キーを指定します。

Example:

```
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50r
v7gsEPjhOBYmsi6PAVKui1nIf/DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQ
W3g9igG30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH3UD/vKyziEh5
4Tplx8=
```

ステップ 6 設定を保存します。

Example:

```
switch(config)# copy running-config startup-config
```

SSH のパスワードが不要なファイル コピーの設定例

次に、Cisco NX-OS デバイスから Secure Copy（SCP）サーバまたは Secure FTP（SFTP）サーバに、パスワードなしでファイルをコピーする例を示します。

Procedure

ステップ 1 SSH の公開キーと秘密キーを生成し、指定したユーザの Cisco NX-OS デバイスのホーム ディレクトリに格納します。

Example:

```
switch# configure terminal
switch(config)# username admin keypair generate rsa
generating rsa key(1024 bits)......
generated rsa key
```

ステップ 2 指定したユーザの公開キーを表示します。

Example:

```
switch(config)# show username admin keypair

**************************************

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZElTfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=

bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
```

ステップ 3 Cisco NX-OS デバイスのホーム ディレクトリから、指定したブートフラッシュ ディレクトリに、公開キーと秘密キーをエクスポートします。

Example:

```
switch(config)# username admin keypair export bootflash:key_rsa rsa
Enter Passphrase:
switch(config)# dir
```

```
.
.
.
        951     Jul 09 11:13:59 2013  key_rsa
        221     Jul 09 11:14:00 2013  key_rsa.pub
.
.
```

ステップ 4 これら 2 つのファイルを他の Cisco NX-OS デバイスへコピーした後、**copy scp** または **copy sftp** コマンドを使用して、Cisco NX-OS デバイスのホーム ディレクトリにインポートします。

Example:

```
switch(config)# username admin keypair import bootflash:key_rsa rsa
Enter Passphrase:
switch(config)# show username admin keypair
**************************************

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZElTfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=

bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
switch(config)#
```

ステップ 5 SCP サーバまたは SFTP サーバで、key_rsa.pub に格納されている公開キーを authorized_keys ファイルに追加します。

Example:

```
$ cat key_rsa.pub >> $HOME/.ssh/ authorized_keys
```

これで、標準の SSH コマンドおよび SCP コマンドを使用してパスワードを指定しなくても、Cisco NX-OS デバイスからサーバにファイルをコピーできます。

ステップ 6 (Optional) DSA キーについてこの手順を繰り返します。

X.509v3 証明書ベースの SSH 認証の設定例

次の例は、X.509v3 証明書を使用する SSH 認証の設定方法を示しています。

（注） Cisco NX-OS リリース 10.4(3)F 以降、Cisco Nexus 9000 シリーズ スイッチは、TACACS+ サーバを介し、X.509 証明書を使用した SSH 認証をサポートしています。この機能は、RADIUS ではサポートされていません。

```
configure terminal
username jsmith password 4Ty18Rnt
username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1,
emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith"
 rsa
crypto ca trustpoint tp1
crypto ca authenticate tp1
crypto ca crl request tp1 bootflash:crl1.crl

show crypto ca certificates
Trustpoint: tp1
CA certificate 0:
subject= /CN=SecDevCA
issuer= /CN=SecDevCA
serial=01AB02CD03EF04GH05IJ06KL07MN
notBefore=Jun 29 12:36:26 2016 GMT
notAfter=Jun 29 12:46:23 2021 GMT
SHA1 Fingerprint=47:29:E3:00:C1:C1:47:F2:56:8B:AC:B2:1C:64:48:FC:F4:8D:53:AF
purposes: sslserver sslclient

show crypto ca crl tp1
Trustpoint: tp1 CRL: Certificate Revocation List (CRL):
    Version 2 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: /CN=SecDevCA
    Last Update: Aug 8 20:03:15 2016 GMT
    Next Update: Aug 16 08:23:15 2016 GMT
    CRL extensions:
        X509v3 Authority Key Identifier:
            keyid:30:43:AA:80:10:FE:72:00:DE:2F:A2:17:E4:61:61:44:CE:78:FF:2A

show user-account
user:user1
        this user account has no expiry date
        roles:network-operator
        ssh cert DN : /C = US, ST = New York, L = Metropolis, O = cisco , OU = csg, CN
= user1; Algo: x509v3-sign-rsa

show users
NAME      LINE        TIME          IDLE      PID         COMMENT
user1     pts/1       Jul 27 18:43  00:03     18796      (10.10.10.1)   session=ssh
```

SSH および Telnet に関する追加情報

ここでは、SSH および Telnet の実装に関する追加情報について説明します。

関連資料

関連項目	マニュアル タイトル
Cisco NX-OS のライセンス	*Cisco NX-OS* ライセンス ガイド
VRF コンフィギュレーション	『*Cisco Nexus 9000* シリーズ *NX-OS* ユニキャスト ルーティング設定ガイド』

RFC

RFC	タイトル
RFC 6187	セキュアシェル認証用の*X.509v3*証明書

MIB

MIB	MIB のリンク
SSH および Telnet に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

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。