SSH および Telnet の設定

この章は、次の項で構成されています。

SSH および Telnet の設定

SSH および Telnet の概要

SSH サーバー

セキュア シェル（SSH）プロトコル サーバー機能を使用すると、SSH クライアントは Cisco Nexus 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 Nexus デバイス スイッチの SSH サーバーは、無償あるいは商用の SSH クライアントと連係して動作します。

SSH がサポートするユーザー認証メカニズムには、RADIUS、TACACS+、およびローカルに格納されたユーザー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 クライアント機能は、SSH プロトコルを介して実行されるアプリケーションで、認証と暗号化を行います。SSH クライアントを使用すると、スイッチは、別の Cisco Nexus デバイス スイッチとの間、または SSH サーバーを稼働している他の任意の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。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 Nexus デバイスの SSH クライアントは、無償あるいは商用の SSH サーバーと連係して動作します。

SSH サーバ キー

SSH では、Cisco Nexus デバイスとのセキュアな通信を行うためにサーバー キーが必要です。SSH 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

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 2 とおりのキー ペアを使用できます。

- dsa オプションを使用すると、SSH バージョン 2 プロトコルに対応する DSA キーペアが生成されます。
- rsa オプションを使用すると、SSH バージョン 2 プロトコルに対応する RSA キーペアが生成されます。

デフォルトでは、Cisco Nexus デバイスは 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）

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。

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 サーバーへのご利用条件を排除し、ラストリゾート 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が向上しました。

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 ID 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 ID を抽出します。
   - ホスト ID 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。
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。必要に応じて `file_name` を交換します。

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 クライアントを構成するための **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 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構成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 を使用するように構成された SSH サーバーを指します。

始める前に

HIBA SSH クライアントを構成する前に、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。
- 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` または同等品）。

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` を編集厳密なホスト キー 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
```

ユーザー公開キーの表示

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 サーバーに接続

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）を活用 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。

正しく構成されている場合、 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
AAAAE2VjZHNhLXNoYTItbmlzdHAzODQAAAAIbmlzdHAzODQAAABhBBPiMs3fwftVJoMIB+HET73Eq0aeTeVBc68DnzEmajZ9EVgkGfEW+0LrbNIy7L/hj5mlsp2WMnlrH+x1qpy1B3a7Y18tERJ4IASIE3JXMgwYYrNZdVI09yK6WjdMLQXCA==
 /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（ID 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ーが別サイトのログイン サーバーとの TCP 接続を確立して、システム間でキーストロークをやり取りできます。Telnet は、リモート システムのアドレスとして、IP アドレスまたはドメイン名を受け取ります。

Cisco Nexus デバイスでは、デフォルトで Telnet サーバーがイネーブルになっています。

SSH の注意事項および制約事項

SSH には、次の注意事項および制限事項があります。

- Cisco Nexus デバイスは、SSH バージョン 2（SSHv2）だけをサポートしています。
- 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 の設定

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。

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}
3. switch(config)# **exit**
4. (Optional) switch# **show ssh key**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	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	switch(config)# **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}	SSH サーバー キーを生成します。

	Command or Action	Purpose
		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、キーワード **force** を使用します。
ステップ 3	switch(config)# **exit**	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) switch# **show ssh key**	SSH サーバー キーを表示します。
ステップ 5	(Optional) 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。

Example

次に、SSH サーバー キーを生成する例を示します。

```
switch# configure terminal
switch(config)# ssh key rsa 2048
switch(config)# exit
switch# show ssh key
switch# copy running-config startup-config
```

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 3 種類のいずれかの形式で指定できます。

- Open SSH 形式
- Internet Engineering Task Force（IETF）SECSH 形式
- Privacy Enhanced Mail（PEM）形式の公開キー証明書

Open SSH 形式による SSH 公開キーの指定

ユーザー アカウント用に SSH 形式で SSH 公開キーを指定できます。

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **username** *username* **sshkey** *ssh-key*
3. switch(config)# **exit**
4. (Optional) switch# **show user-account**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	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	switch(config)# **username** *username* **sshkey** *ssh-key*	SSH 形式で SSH 公開キーを設定します。
ステップ 3	switch(config)# **exit**	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) switch# **show user-account**	ユーザー アカウントの設定を表示します。
ステップ 5	(Optional) 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。

Example

次に、Open SSH 形式で SSH 公開キーを指定する例を示します。

```
switch# configure terminal
switch(config)# username User1 sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYz
CfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZcTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4Z
XIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5Ninn0McNinn0Mc=
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

Note 上記の例の **username** コマンドは、読みやすくするために改行されていますが、単一行です。

IETF SECSH 形式による SSH 公開キーの指定

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

SUMMARY STEPS

1. switch# **copy** *server-file* **bootflash:** *filename*
2. switch# **configure terminal**
3. switch(config)# **username** *username* **sshkey file** *filename*
4. switch(config)# **exit**
5. (Optional) switch# **show user-account**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	switch# **copy** *server-file* **bootflash:** *filename*	サーバーから 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。File Transfer Protocol（FTP）、SCP、SSH File Transfer Protocol（SFTP）、または Trivial File Transfer Protocol（TFTP）サーバーを利用できます。
ステップ 2	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 3	switch(config)# **username** *username* **sshkey file** *filename*	SSH 形式で SSH 公開キーを設定します。
ステップ 4	switch(config)# **exit**	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) switch# **show user-account**	ユーザー アカウントの設定を表示します。
ステップ 6	(Optional) 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。

Example

次に、IETF SECSH 形式で SSH 公開キーを指定する例を示します。

```
switch#copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub
switch# configure terminal
switch(config)# username User1 sshkey file bootflash:secsh_file.pub
switch(config)# exit
switch# show user-account
switch# copy running-config startup-config
```

PEM フォーマット化された公開キー証明書形式による SSH 公開キーの指定

ユーザー アカウント用に PEM フォーマット化された公開キー証明書形式で SSH 公開キーを指定できます。

SUMMARY STEPS

1. switch# **copy** *server-file* **bootflash:** *filename*
2. switch# **configure terminal**
3. (Optional) switch# **show user-account**
4. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	switch# **copy** *server-file* **bootflash:** *filename*	サーバーから PEM フォーマット化された公開キー証明書形式の SSH キーを含むファイルをダウンロードします。FTP、SCP、SFTP、または TFTP サーバーを利用できます。
ステップ 2	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 3	(Optional) switch# **show user-account**	ユーザー アカウントの設定を表示します。
ステップ 4	(Optional) 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。

Example

次に、PEM フォーマット化された公開キー証明書形式で SSH 公開キーを指定する例を示します。

```
switch# copy tftp://10.10.1.1/cert.pem bootflash:cert.pem
switch# configure terminal
switch# show user-account
switch# copy running-config startup-config
```

リモート デバイスとの SSH セッションの開始

Cisco Nexus デバイスからリモート デバイスに接続する SSH セッションを開始できます。

SUMMARY STEPS

1. switch# **ssh** {*hostname* | *username@hostname*} [**vrf** *vrf-name*]

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	switch# **ssh** {*hostname* | *username@hostname*} [**vrf** *vrf-name*]	リモート デバイスとの SSH セッションを作成します。引数 *hostname* には、IPv4 アドレスまたはホスト名を指定します。

SSH ホストのクリア

SCP または SFTP を使用してサーバーからファイルをダウンロードする場合は、サーバーと信頼性のある SSH 関係を確立します。

SUMMARY STEPS

1. switch# **clear ssh hosts**

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	switch# **clear ssh hosts**	SSH ホスト セッションをクリアします。

SSH サーバのディセーブル化

SSH サーバーは、デフォルトでCisco Nexus デバイスでイネーブルになっています。

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **[no] feature ssh**
3. switch(config)# **exit**
4. (Optional) switch# **show ssh server**
5. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	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	switch(config)# **[no] feature ssh**	SSH サーバーをイネーブル/ディセーブルにします。デフォルトでは有効になっています。
ステップ 3	switch(config)# **exit**	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) switch# **show ssh server**	SSH サーバーの設定を表示します。
ステップ 5	(Optional) 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 サーバーをディセーブルにした後、SSH サーバー 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 サーバー キーを生成する必要があります。

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no feature ssh**
3. switch(config)# **no ssh key** [**dsa** | **rsa**]
4. switch(config)# **exit**
5. (Optional) switch# **show ssh key**
6. (Optional) switch# **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	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	switch(config)# **no feature ssh**	SSH サーバーをディセーブルにします。
ステップ 3	switch(config)# **no ssh key** [**dsa** | **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	switch(config)# **exit**	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) switch# **show ssh key**	SSH サーバーの設定を表示します。
ステップ 6	(Optional) 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 セッションのクリア

Cisco Nexus デバイスから SSH セッションをクリアできます。

SUMMARY STEPS

1. switch# **show users**
2. switch# **clear line** *vty-line*

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	switch# **show users**	ユーザー セッション情報を表示します。
ステップ 2	switch# **clear line** *vty-line*	ユーザ SSH セッションをクリアします。

SSH の設定例

次に、SSH を設定する例を示します。

SUMMARY STEPS

1. SSH サーバ キーを生成します。
2. SSH サーバをイネーブルにします。
3. SSH サーバー キーを表示します。
4. Open SSH 形式による SSH 公開キーを指定します。
5. 設定を保存します。

DETAILED STEPS

Procedure

ステップ 1 SSH サーバ キーを生成します。

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

ステップ 2 SSH サーバをイネーブルにします。

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

Note
SSH サーバーはデフォルトでイネーブルになっているため、この手順は必要ありません。

ステップ 3 SSH サーバー キーを表示します。

```
switch(config)# show ssh key
rsa Keys generated:Fri May  8 22:09:47 2009
```

```
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYzCfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZ/
cTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4ZXIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5/
Ninn0Mc=

bitcount:1024
fingerprint:
4b:4d:f6:b9:42:e9:d9:71:3c:bd:09:94:4a:93:ac:ca
***********************************
could not retrieve dsa key information
***********************************
```

ステップ 4 Open SSH 形式による SSH 公開キーを指定します。

```
switch(config)# username User1 sshkey ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAri3mQy4W1AV9Y2t2hrEWgbUEYz
CfTPO5B8LRkedn56BEy2N9ZcdpqE6aqJLZwfZcTFEzaAAZp9AS86dgBAjsKGs7UxnhGySr8ZELv+DQBsDQH6rZt0KR+2Da8hJD4Z
XIeccWk0gS1DQUNZ300xstQsYZUtqnx1bvm5Ninn0McNinn0Mc=
```

ステップ 5 設定を保存します。

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

Telnet の設定

Telnet サーバのイネーブル化

デフォルトでは、Telnet サーバーはイネーブルに設定されています。Cisco Nexus デバイスの Telnet サーバーをディセーブルにできます。

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **[no] feature telnet**

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	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。

	Command or Action	Purpose
ステップ 2	switch(config)# **[no] feature telnet**	Telnet サーバーをイネーブル/ディセーブルにします。デフォルトではイネーブルになっています。

Telnet サーバーの再イネーブル化

Cisco Nexus デバイスの Telnet サーバーがディセーブルにされた場合は、再度イネーブルにできます。

SUMMARY STEPS

1. switch(config)# **[no] feature telnet**

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	switch(config)# **[no] feature telnet**	Telnet サーバーを再度イネーブルにします。

リモート デバイスとの Telnet セッションの開始

Telnetセッションを開始してリモートデバイスに接続する前に、次の作業を行う必要があります。

- リモートデバイスのホスト名を取得します。必要に応じて、リモートデバイスのユーザー名も取得します。
- Cisco Nexus デバイス上で Telnet サーバーをイネーブルにします。
- リモート デバイス上で Telnet サーバーをイネーブルにします。

SUMMARY STEPS

1. switch# **telnet** *hostname*

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	switch# **telnet** *hostname*	リモートデバイスとのTelnetセッションを作成します。引数 *hostname* には、IPv4 アドレスまたはデバイス名を指定します。

Example

次に、Telnet セッションを開始してリモート デバイスに接続する例を示します。

```
switch# telnet 10.10.1.1
Trying 10.10.1.1...
Connected to 10.10.1.1.
Escape character is '^]'.
switch login:
```

Telnet セッションのクリア

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

SUMMARY STEPS

1. switch# **show users**
2. switch# **clear line** *vty-line*

DETAILED STEPS

Procedure

	Command or Action	Purpose
ステップ 1	switch# **show users**	ユーザー セッション情報を表示します。
ステップ 2	switch# **clear line** *vty-line*	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 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。

Procedure

- switch# **show ssh key** [**dsa** | **rsa**]

コマンドまたはアクション	目的
switch# **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 およびユーザー アカウントのデフォルト値が表示されます。
switch# **show ssh server**	SSH サーバーの設定を表示します。
switch# **show user-account**	ユーザー アカウント情報を表示します。

SSH のデフォルト設定

次の表に、SSH 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* 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
Telnet サーバ	有効（Enabled）

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。