Configuring SSH and Telnet

This chapter contains the following sections:

Configuring SSH and Telnet

Information About SSH and Telnet

SSH Server

The Secure Shell Protocol (SSH) server feature enables a SSH client to make a secure, encrypted connection to a Cisco Nexus device. SSH uses strong encryption for authentication. The SSH server in the Cisco Nexus device switch interoperates with publicly and commercially available SSH clients.

The user authentication mechanisms supported for SSH are RADIUS, TACACS+, and the use of locally stored user names and passwords.

SSH Client

The SSH client feature is an application running over the SSH protocol to provide device authentication and encryption. The SSH client enables a switch to make a secure, encrypted connection to another Cisco Nexus device or to any other device running an SSH server. This connection provides an outbound connection that is encrypted. With authentication and encryption, the SSH client allows for a secure communication over an insecure network.

The SSH client in the Cisco Nexus device works with publicly and commercially available SSH servers.

SSH Server Keys

SSH requires server keys for secure communications to the Cisco Nexus device. You can use SSH keys for the following SSH options:

- SSH version 2 using Rivest, Shamir, and Adelman (RSA) public-key cryptography
- SSH version 2 using the Digital System Algrorithm (DSA)

Be sure to have an SSH server key-pair with the appropriate version before enabling the SSH service. You can generate the SSH server key-pair according to the SSH client version used. The SSH service accepts three types of key-pairs for use by SSH version 2:

- The dsa option generates the DSA key-pair for the SSH version 2 protocol.
- The rsa option generates the RSA key-pair for the SSH version 2 protocol.

By default, the Cisco Nexus device generates an RSA key using 1024 bits.

SSH supports the following public key formats:

- OpenSSH
- IETF Secure Shell (SECSH)

Caution If you delete all of the SSH keys, you cannot start the SSH services.

SSH Authentication Using Host Identity Based Authorization (HIBA)

Host-Based Authentication is an SSH authentication method that authenticates the client's host to the server (Cisco Nexus 9000 switch) by verifying the client's host public key in the server's `known_hosts` file. This is distinct from SSH Certificate-Based Authentication, which uses digital certificates signed by a Certificate Authority (CA) to authenticate users or hosts.

Host Identity Based Authorization (HIBA) is a method that centralizes SSH authorization management by embedding host authorization information within certificates.

- Host authorization information is embedded in the host certificate.
- User certificates contain "grants" specifying permitted access.
- Authorization is managed centrally by a Certificate Authority (CA).

HIBA simplifies SSH access control, reduces administrative overhead, and eliminates dependencies on external AAA servers for authorization.

Benefits of HIBA

HIBA offers several advantages over traditional SSH key management:

Key benefits of HIBA include:

- **Simplified Management:** Centralized authorization through certificate-based identity simplifies management.
- **Scalability:** Easier management of SSH access in large, complex environments.
- **Reduced Dependencies:** Eliminates the dependency on external AAA servers for authorization, making it suitable for last-resort access.
- **Enhanced Security:** Improves control over temporary and privileged access with short-lived certificates.

How SSH Authentication with HIBA Works

This process describes how SSH authentication occurs when HIBA is configured.

Summary

The SSH server invokes the HIBA authorization module to process user certificates during authentication. Access is granted if the HIBA module successfully validates the user's certificate against the configured host identity and grants. If HIBA validation fails, the SSH server may fall back to other authentication methods, depending on the configuration.

Workflow

These stages describe the SSH authentication process with HIBA:

1. **SSH Connection Attempt** - A user attempts to connect to the switch via SSH.
2. **Certificate Presentation** - The SSH client presents the user's certificate to the SSH server on the switch.
3. **HIBA Module Invocation** - The SSH server, based on its configuration (AuthorizedPrincipalsCommand), invokes the HIBA authorization module.
4. **Certificate Validation** - The HIBA module performs the following validations:
   - Verifies the user certificate's signature against the configured HIBA CA.
   - Extracts the host identity from the host certificate.
   - Checks for a valid "grant" in the user certificate that matches the host identity.
5. **Access Decision** - Based on the HIBA module's validation, one of the following occurs:

When...	And...	Then...	And...
The user certificate is successfully validated by the HIBA module	A valid grant for the target host is found in the user certificate	Access is granted to the user.	The SSH session proceeds.
The user certificate is invalid or cannot be validated.	No valid grant is found in the user certificate.	Access is denied by the HIBA module.	The SSH server may fall back to other authentication methods (if configured).

Configuring HIBA for SSH Authentication

This steps guides you through the configuration of SSH Host Identity Based Authorization (HIBA).

This configuration involves generating SSH server keys, configuring a trustpoint for the HIBA CA, enrolling the SSH host certificate, and configuring the SSH server to use HIBA for authentication.

Note To configure HIBA for the first time, you can log in to the switch using traditional SSH authentication methods, such as local user accounts or other configured AAA servers. Enabling HIBA does not remove or block existing local SSH users unless you explicitly delete those accounts.

Before you begin

Before configuring HIBA, ensure that you have:

- A functional PKI infrastructure, including a Certificate Authority (CA).
- Connectivity to the CA server.

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
```

Enter global configuration mode.

Step 2 **ssh key ecdsa** *bits*

Example:

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

Generate ECDSA keypair for the switch. This example uses a 384-bit ECDSA key. Use a key size supported by your security policy and platform.

Step 3 **ssh key export** *bootflash:file_name* **ecdsa**

Example:

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

Export the SSH host ECDSA key to bootflash. Replace `file_name` as needed.

After export, transfer `host_key` and `host_key.pub` files to your CA machine using SFTP:

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

Step 4 **crypto ca trustpoint** *openssh-ca* **type ssh**

Example:

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

Create a trustpoint for the HIBA CA. Use the name **openssh-ca** for consistency.

Step 5 **crypto ca authenticate** *openssh-ca* **type ssh** *ecdsa-sha2-nistp384 public_key*

Example:

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

Authenticate the HIBA CA by importing the CA public key. Replace the key string with your actual CA public key.

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

Example:

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

Enroll the SSH host certificate signed by your CA. Use the certificate content generated as per the Google HIBA CA wiki instructions.

Example Configuration: HIBA SSH Client on Linux

Important The following steps are provided as an **example** for configuring a HIBA SSH client on a Linux system. The exact procedure and output may vary depending on your client operating system and SSH version. Consult your system's official SSH documentation for authoritative instructions.

This steps guides you through the client-side configuration for using Host Identity Based Authorization (HIBA) with SSH.

Note The term "HIBA server" refers to the SSH server running on the Cisco Nexus 9000 switch, configured to use HIBA.

Before you begin

Before configuring the HIBA SSH client, ensure that you have:

- A valid installation of `openssh-client` on your host.
- The CA public key (`ca.pub`).
- Your user private key and matching certificate with a valid HIBA extension.
- Your user public key (`key_rsa.pub` or equivalent).

Procedure

Step 1 **$ cat /etc/ssh/ssh_config**

Example:

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

Configure SSH client settings

Edit `/etc/ssh/ssh_config` to enable strict host key checking and specify a `GlobalKnownHostsFile` that will contain your CA public key for SSH certificate validation.

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

Example:

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

Populate **known_hosts** with CA public key

Add the CA public key to the `known_hosts` file using the `@cert-authority` directive. This step ensures the SSH client trusts any host certificate signed by this CA.

Step 3 **$ cat ~/.ssh/key_rsa.pub**

Example:

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

View your user public key

Display the contents of your user public key file. This key is required for certificate-based authentication and should correspond to your private key and certificate.

Note
If your key has a different name or location, adjust the path accordingly.

Step 4 **$ ssh -i <path_to_private_key> <user>@<hiba_server_ip>**

Example:

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

Connect to the HIBA-enabled SSH server

Use your private key (and its matching certificate, if required) to connect to the SSH server.

Note
The `-i` option specifies the user's private key (identity file).

If configured correctly, the SSH connection should be established using HIBA certificate-based authentication, and host validation will succeed against the CA public key. Passwordless login will be possible if the public key is present in `authorized_keys` on the server.

Verifying HIBA Configuration

Procedure

Step 1 **show crypto ca certificates type ssh**

Example:

```
switch(config)# show crypto ca certificates type ssh
            trustpoint: openssh-ca
            CA Public Key:
            ecdsa-sha2-nistp384
AAAAE2VjZHNhLXNoYTItbmlzdHAzODQAAAAIbmlzdHAzODQAAABhBBPiMs3fwftVUoMIB+HET73Eqp0aeTeVBo68DnzEmajZ9EVgkGfEW+0LnbNIy7L/hj5mlsp2VMnlrHx1qay1B3a7Y18tERJ4IASIE3JXMgwYYrNNZdVI09yK6WjdMLQXCA==
 /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'
```

Display the SSH certificates and verify that the host certificate is enrolled and associated with the correct trustpoint (`openssh-ca`).

Expected Output: The output should display the SSH host certificate details, including the "HIBA Info" section, which shows the HIBA grants.

If the host certificate and HIBA information are displayed correctly, the certificate enrollment is successful.

Step 2 **show crypto ca trustpoints type ssh**

Example:

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

Display the SSH trustpoints and verify that the HIBA CA trustpoint (`openssh-ca`) is present.

Expected Output: The output should list the trustpoint names of type `ssh`.

If the HIBA CA trustpoint appears in the output, the trustpoint has been configured successfully.

Step 3 **ssh -i path_to_private_key> <user>@<switch_ip**

Example:

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

Attempt to SSH to the switch using a user with a HIBA-enabled certificate signed by the CA.

Note: The **-i** option specifies the path to the user's `private key` (identity file). The HIBA extension must be included in the certificate that pairs with this private key, and the CA public key must be trusted by the switch. Ensure the private key file is kept secure.

The SSH connection should be established successfully without prompting for a password (if password authentication is disabled).

Telnet Server

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site, and then passes the keystrokes from one system to the other. Telnet can accept either an IP address or a domain name as the remote system address.

The Telnet server is enabled by default on the Cisco Nexus device.

Guidelines and Limitations for SSH

SSH has the following configuration guidelines and limitations:

- The Cisco Nexus device supports only SSH version 2 (SSHv2).
- SSH public and private keys imported into user accounts that are remotely authenticated through a AAA protocol (such as RADIUS or TACACS+) for the purpose of SSH Passwordless File Copy will not persist when the Nexus device is reloaded unless a local user account with the same name as the remote user account is configured on the device before the SSH keys are imported.

Configuring SSH

Generating SSH Server Keys

You can generate an SSH server key based on your security requirements. The default SSH server key is an RSA key that is generated using 1024 bits.

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}	Generates the SSH server key. The *bits* argument is the number of bits used to generate the key. The range is from 768 to 4096 and the default value is 1024. Use the **force** keyword to replace an existing key.
Step 3	switch(config)# **exit**	Exits global configuration mode.

	Command or Action	Purpose
Step 4	(Optional) switch# **show ssh key**	Displays the SSH server keys.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to generate an SSH server key:

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

Specifying the SSH Public Keys for User Accounts

You can configure an SSH public key to log in using an SSH client without being prompted for a password. You can specify the SSH public key in one of three different formats:

- Open SSH format
- IETF SECSH format
- Public Key Certificate in PEM format

Specifying the SSH Public Keys in Open SSH Format

You can specify the SSH public keys in SSH format for user accounts.

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **username** *username* **sshkey** *ssh-key*	Configures the SSH public key in SSH format.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show user-account**	Displays the user account configuration.

	Command or Action	Purpose
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to specify an SSH public key in open SSH format:

```
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 The **username** command in the example above is a single line that has been broken for legibility.

Specifying the SSH Public Keys in IETF SECSH Format

You can specify the SSH public keys in IETF SECSH format for user accounts.

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
Step 1	switch# **copy** *server-file* **bootflash:** *filename*	Downloads the file that contains the SSH key in IETF SECSH format from a server. The server can be FTP, SCP, SFTP, or TFTP.
Step 2	switch# **configure terminal**	Enters global configuration mode.
Step 3	switch(config)# **username** *username* **sshkey file** *filename*	Configures the SSH public key in SSH format.
Step 4	switch(config)# **exit**	Exits global configuration mode.

	Command or Action	Purpose
Step 5	(Optional) switch# **show user-account**	Displays the user account configuration.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to specify the SSH public key in the IETF SECSH format:

```
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
```

Specifying the SSH Public Keys in PEM-Formatted Public Key Certificate Form

You can specify the SSH public keys in PEM-formatted Public Key Certificate form for user accounts.

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
Step 1	switch# **copy** *server-file* **bootflash:** *filename*	Downloads the file that contains the SSH key in PEM-formatted Public Key Certificate form from a server. The server can be FTP, SCP, SFTP, or TFTP
Step 2	switch# **configure terminal**	Enters global configuration mode.
Step 3	(Optional) switch# **show user-account**	Displays the user account configuration.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to specify the SSH public keys in PEM-formatted public key certificate form:

```
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
```

Starting SSH Sessions to Remote Devices

You can start SSH sessions to connect to remote devices from your Cisco Nexus device.

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **ssh** {*hostname* | *username@hostname*} [**vrf** *vrf-name*]	Creates an SSH session to a remote device. The *hostname* argument can be an IPv4 address or a hostname.

Clearing SSH Hosts

When you download a file from a server using SCP or SFTP, you establish a trusted SSH relationship with that server.

SUMMARY STEPS

1. switch# **clear ssh hosts**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **clear ssh hosts**	Clears the SSH host sessions.

Disabling the SSH Server

By default, the SSH server is enabled on the Cisco Nexus device.

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] feature ssh**	Enables/disables the SSH server. The default is enabled.
Step 3	switch(config)# **exit**	Exits global configuration mode.
Step 4	(Optional) switch# **show ssh server**	Displays the SSH server configuration.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Deleting SSH Server Keys

You can delete SSH server keys after you disable the SSH server.

Note To reenable SSH, you must first generate an SSH server key.

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature ssh**	Disables the SSH server.
Step 3	switch(config)# **no ssh key** [**dsa** | **rsa**]	Deletes the SSH server key. The default is to delete all the SSH keys.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	(Optional) switch# **show ssh key**	Displays the SSH server configuration.

	Command or Action	Purpose
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Clearing SSH Sessions

You can clear SSH sessions from the Cisco Nexus device.

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **show users**	Displays user session information.
Step 2	switch# **clear line** *vty-line*	Clears a user SSH session.

Configuration Examples for SSH

The following example shows how to configure SSH:

SUMMARY STEPS

1. Generate an SSH server key.
2. Enable the SSH server.
3. Display the SSH server key.
4. Specify the SSH public key in Open SSH format.
5. Save the configuration.

DETAILED STEPS

Procedure

Step 1 Generate an SSH server key.

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

Step 2 Enable the SSH server.

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

Note
This step should not be required because the SSH server is enabled by default.

Step 3 Display the SSH server key.

```
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
***********************************
```

Step 4 Specify the SSH public key in Open SSH format.

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

Step 5 Save the configuration.

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

Configuring Telnet

Enabling the Telnet Server

By default, the Telnet server is enabled. You can disable the Telnet server on your Cisco Nexus device.

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] feature telnet**	Enables/disables the Telnet server. The default is enabled.

Reenabling the Telnet Server

If the Telnet server on your Cisco Nexus device has been disabled, you can reenable it.

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **[no] feature telnet**	Reenables the Telnet server.

Starting Telnet Sessions to Remote Devices

Before you start a Telnet session to connect to remote devices, you should do the following:

- Obtain the hostname for the remote device and, if needed, obtain the username on the remote device.
- Enable the Telnet server on the Cisco Nexus device.
- Enable the Telnet server on the remote device.

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **telnet** *hostname*	Creates a Telnet session to a remote device. The *hostname* argument can be an IPv4 address or a device name.

Example

The following example shows how to start a Telnet session to connect to a remote device:

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

Clearing Telnet Sessions

You can clear Telnet sessions from the Cisco Nexus device.

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **show users**	Displays user session information.
Step 2	switch# **clear line** *vty-line*	Clears a user Telnet session.

Verifying the SSH and Telnet Configuration

To display the SSH configuration information, perform one of the following tasks:

Procedure

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

Command or Action	Purpose
switch# **show running-config security**[**all**]	Displays the SSH and user account configuration in the running configuration. The **all**keyword displays the default values for the SSH and user accounts.
switch# **show ssh server**	Displays the SSH server configuration.
switch# **show user-account**	Displays user account information

Default Settings for SSH

The following table lists the default settings for SSH parameters.

Table 1: Default SSH Parameters

Parameters	Default
SSH server	Enabled
SSH server key	RSA key generated with 1024 bits
RSA key bits for generation	1024
Telnet server	Enabled