Configuring SSH and Telnet

This chapter describes how to configure Secure Shell Protocol (SSH) and Telnet on Cisco NX-OS devices.

This chapter includes the following sections:

About SSH and Telnet

This section includes information about SSH and Telnet.

SSH Server

You can use the SSH server to enable an SSH client to make a secure, encrypted connection to a Cisco NX-OS device. SSH uses strong encryption for authentication. The SSH server in the Cisco NX-OS software can interoperate with publicly and commercially available SSH clients.

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

SSH Client

The SSH client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco NX-OS device to make a secure, encrypted connection to another Cisco NX-OS device or to any other device that runs the 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 NX-OS software works with publicly and commercially available SSH servers.

SSH Server Keys

SSH requires server keys for secure communications to the Cisco NX-OS device. You can use SSH server 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)
- SSH version 2 using the Elliptic Curve Digital Signature Algorithm (ECDSA)

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 two 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.
- The **ecdsa** option generates the ECDSA key-pair for the SSH version 2 protocol.

By default, the Cisco NX-OS software generates an RSA key using 1024 bits.

SSH supports the following public key formats:

- OpenSSH
- IETF Secure Shell (SECSH)
- Public Key Certificate in Privacy-Enhanced Mail (PEM)

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

SSH Authentication Using Digital Certificates

SSH authentication on Cisco NX-OS devices provide X.509 digital certificate support for host authentication. An X.509 digital certificate is a data item that ensures the origin and integrity of a message. It contains encryption keys for secured communications and is signed by a trusted certification authority (CA) to verify the identity of the presenter. The X.509 digital certificate support provides either DSA or RSA algorithms for authentication.

The certificate infrastructure uses the first certificate that supports the Secure Socket Layer (SSL) and is returned by the security infrastructure, either through a query or a notification. Verification of certificates is successful if the certificates are from any of the trusted CAs configured and if not revoked or expired.

You can configure your device for SSH authentication using an X.509 certificate. If the authentication fails, you are prompted for a password.

You can configure SSH authentication using X.509v3 certificates (RFC 6187). X.509v3 certificate-based SSH authentication uses certificates combined with a smartcard to enable two-factor authentication for Cisco device access. The SSH client is provided by Cisco partner Pragma Systems.

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 device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

The Telnet server is disabled by default on the Cisco NX-OS device.

Licensing Requirements for SSH and Telnet

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	SSH and Telnet require no license. Any feature not included in a license package is bundled with the nx-os image and is provided at no extra charge to you. For an explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for SSH and Telnet

Make sure that you have configured IP on a Layer 3 interface, out-of-band on the mgmt 0 interface, or inband on an Ethernet interface.

Guidelines and Limitations for SSH and Telnet

SSH and Telnet have the following configuration guidelines and limitations:

- The Cisco NX-OS software supports only SSH version 2 (SSHv2).
- Due to a Poodle vulnerability, SSLv3 is no longer supported.
- You can configure your device for SSH authentication using an X.509 certificate. If the authentication fails, you are prompted for a password.
- The SFTP server feature does not support the regular SFTP **chown** and **chgrp** commands.
- When the SFTP server is enabled, only the admin user can use SFTP to access the device.
- 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.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings for SSH and Telnet

This table lists the default settings for SSH and Telnet parameters.

Table 1: Default SSH and Telnet Parameters

Parameters	Default
SSH server	Enabled
SSH server key	RSA key generated with 1024 bits
RSA key bits for generation	1024
Telnet server	Disabled
Telnet port number	23
Maximum number of SSH login attempts	3
SCP server	Disabled
SFTP server	Disabled

Configuring SSH

This section describes how to configure 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.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.
Step 3	**ssh key** {**dsa** [**force**] | **rsa** [*bits*[**force**]] | **ecdsa** [*bits* [**force**]]} **Example:** `switch(config)# ssh key rsa 2048`	Generates the SSH server key. The *bits* argument is the number of bits used to generate the RSA key. The range is from 768 to 2048. The default value is 1024. You cannot specify the size of the DSA key. It is always set to 1024 bits. Use the **force** keyword to replace an existing key. **Note** If you configure ssh key dsa, you must do the following additional configurations: ssh keytypes all and ssh kexalgos all
Step 4	**ssh rekey max-data** *max-data* **max-time** *max-time*i **Example:** `switch(config)# ssh rekey max-data 1K max-time 1M`	Configures the rekey parameters.
Step 5	**feature ssh** **Example:** `switch(config)# feature ssh`	Enables SSH.

Step 6	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 7	(Optional) **show ssh key** [**dsa** | **rsa** | **ecdsa**] [] **Example:** `switch# show ssh key`	Displays the SSH server keys. This command displays the fingerprint in SHA256 format by default. SHA256 is more secure than the old default format of MD5. However, the **md5** option has been added, if you want to see the fingerprint in MD5 format for backward compatibility.
Step 8	**show run security all**	
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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 these formats:

- OpenSSH format
- IETF SECSH format

Specifying the SSH Public Keys in IETF SECSH Format

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

Before you begin

Generate an SSH public key in IETF SCHSH format.

Procedure

	Command or Action	Purpose
Step 1	**copy** *server-file* **bootflash:***filename* **Example:** `switch# copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub`	Downloads the file containing the SSH key in IETF SECSH format from a server. The server can be FTP, secure copy (SCP), secure FTP (SFTP), or TFTP.
Step 2	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.
Step 3	**username** *username* **sshkey file bootflash:***filename* **Example:** `switch(config)# username User1 sshkey file bootflash:secsh_file.pub`	Configures the SSH public key in IETF SECSH format.
Step 4	**exit** **Example:** `switch(config)# exit switch#`	Exits global configuration mode.
Step 5	(Optional) **show user-account** **Example:** `switch# show user-account`	Displays the user account configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Specifying the SSH Public Keys in OpenSSH Format

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

Before you begin

Generate an SSH public key in OpenSSH format.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**username** *username* **sshkey** *ssh-key*
	Example:
	`switch(config)# username User1 sshkey ssh-rsa`
	`AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QeZ19G+3flXswKOiWH7yUjuA50rv7gsEPj`
	`hOBfmsi6PAVkuilnIf/DQhum+lJNqP/eLowb7ibO+lVkRXFY/GHlNIQW6g9igG3Oc6K6+`
	`XVn+NjnIlB7ihvpVh7dLddMOXwOnXHYshXmGiH3UD/vKyziEh5S4Tplx8=`
		Configures the SSH public key in OpenSSH format.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show user-account** **Example:** `switch# show user-account`	Displays the user account configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Maximum Number of SSH Login Attempts

You can configure the maximum number of SSH login attempts. If the user exceeds the maximum number of permitted attempts, the session disconnects.

Note The total number of login attempts includes attempts through public-key authentication, certificate-based authentication, and password-based authentication. If public-key authentication is enabled, it takes priority. If only certificate-based and password-based authentication are enabled, certificate-based authentication takes priority. If you exceed the configured number of login attempts through all of these methods, a message appears indicating that too many authentication failures have occurred.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**ssh login-attempts** *number* **Example:** `switch(config)# ssh login-attempts 5`	Configures the maximum number of times that a user can attempt to log into an SSH session. The default maximum number of login attempts is 3. The range is from 1 to 10. **Note** The **no** form of this command removes the previous login attempts value and sets the maximum number of login attempts to the default value of 3.
Step 3	(Optional) **show running-config security all** **Example:** `switch(config)# show running-config security all`	Displays the configured maximum number of SSH login attempts.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Starting SSH Sessions

You can start SSH sessions using IPv4 or IPv6 to connect to remote devices from the Cisco NX-OS device.

Before you begin

Obtain the hostname for the remote device and, if needed, the username on the remote device.

Enable the SSH server on the remote device.

Procedure

	Command or Action	Purpose
Step 1	**ssh** [*username@*]{*ipv4-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `switch# ssh 10.10.1.1`	Creates an SSH IPv4 session to a remote device using IPv4. The default VRF is the default VRF.
Step 2	**ssh6** [*username@*]{*ipv6-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `switch# ssh6 HostA`	Creates an SSH IPv6 session to a remote device using IPv6.

Starting SSH Sessions from Boot Mode

You can start SSH sessions from the boot mode of the Cisco NX-OS device to connect to remote devices.

Before you begin

Obtain the hostname for the remote device and, if needed, the username on the remote device.

Enable the SSH server on the remote device.

Procedure

	Command or Action	Purpose
Step 1	**ssh** [*username@*]*hostname* **Example:** `switch(boot)# ssh user1@10.10.1.1`	Creates an SSH session to a remote device from the boot mode of the Cisco NX-OS device. The default VRF is always used.
Step 2	**exit** **Example:** `switch(boot)# exit`	Exits boot mode.
Step 3	**copy scp:**//[*username@*]*hostname*/*filepath directory* **Example:** `switch# copy scp://user1@10.10.1.1/users abc`	Copies a file from the Cisco NX-OS device to a remote device using the Secure Copy Protocol (SCP). The default VRF is always used.

Configuration Example for SSH Passwordless File Copy

The following example shows how to copy files from a Cisco NX-OS device to a secure copy (SCP) or secure FTP (SFTP) server without a password:

Procedure

Step 1 Generate the SSH public and private keys and store them in the home directory of the Cisco NX-OS device for the specified user.

Example:

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

Step 2 Display the public key for the specified user.

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

Step 3 Export the public and private keys from the home directory of the Cisco NX-OS device to the specified bootflash directory.

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
.
.
```

Step 4 After copying these two files to another Cisco NX-OS device using the **copy scp** or **copy sftp** command, import them to the home directory of the Cisco NX-OS device.

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+oQhIcvnrMbx2BmD0P8boZEl1TfJ
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)#
```

Step 5 On the SCP or SFTP server, append the public key stored in key_rsa.pub to the authorized_keys file.

Example:

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

You can now copy files from the Cisco NX-OS device to the server without a password using standard SSH and SCP commands.

Step 6 (Optional) Repeat this procedure for the DSA keys.

Configuring SCP and SFTP Servers

You can configure an SCP or SFTP server on the Cisco NX-OS device in order to copy files to and from a remote device. After you enable the SCP or SFTP server, you can execute an SCP or SFTP command on the remote device to copy the files to or from the Cisco NX-OS device.

Note The arcfour and blowfish cipher options are not supported for the SCP server.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature scp-server** **Example:**	Enables or disables the SCP server on the Cisco NX-OS device.

	Command or Action	Purpose
	`switch(config)# feature scp-server`	
Step 3	Required: [**no**] **feature sftp-server** **Example:** `switch(config)# feature sftp-server`	Enables or disables the SFTP server on the Cisco NX-OS device.

Step 4	Required: **exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 5	(Optional) **show running-config security** **Example:** `switch# show running-config security`	Displays the configuration status of the SCP and SFTP servers.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring X.509v3 Certificate-Based SSH Authentication

You can configure SSH authentication using X.509v3 certificates.

Before you begin

Enable the SSH server on the remote device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**username** *user-id* [**password** [**0** | **5**] *password*] **Example:** `switch(config)# username jsmith password 4Ty18Rnt`	Configures a user account. The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 28 characters. Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hyphen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames but not in local usernames.

	Command or Action	Purpose
		Usernames must begin with an alphanumeric character. The default password is undefined. The **0** option indicates that the password is clear text, and the **5** option indicates that the password is encrypted. The default is **0** (clear text). **Note** If you do not specify a password, the user might not be able to log in to the Cisco NX-OS device. **Note** If you create a user account with the encrypted password option, the corresponding SNMP user will not be created.
Step 3	**username** *user-id* **ssh-cert-dn** *dn-name* {**dsa** | **rsa**} **Example:** `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`	Specifies an SSH X.509 certificate distinguished name and DSA or RSA algorithm to use for authentication for an existing user account. The distinguished name can be up to 512 characters and must follow the format shown in the examples. Make sure the email address and state are configured as emailAddress and ST, respectively.

Step 4	[**no**] **crypto ca trustpoint** *trustpoint*
	Example:
	`switch(config)# crypto ca trustpoint winca`
	`switch(config-trustpoint)#`
		Configures a trustpoint.
		Note Before you delete a trustpoint using the **no** form of this command, you must first delete the CRL and CA certificate, using the **delete crl** and **delete ca-certificate** commands.

Step 5	**crypto ca authenticate** *trustpoint* **Example:** `switch(config-trustpoint)# crypto ca authenticate winca`	Configures a CA certificate for the trustpoint. **Note** To delete a CA certificate, enter the **delete ca-certificate** command in the trustpoint configuration mode.
Step 6	(Optional) **crypto ca crl request** *trustpoint* **bootflash:***static-crl***.crl** **Example:** `switch(config-trustpoint)# crypto ca crl request winca bootflash:crllist.crl`	This command is optional but highly recommended. Configures the certificate revocation list (CRL) for the trustpoint. The CRL file is a snapshot of the list of revoked certificates by the trustpoint. This static CRL list is manually copied to the device from the Certification Authority (CA). **Note** Static CRL is the only supported revocation check method.

	Command or Action	Purpose
		Note To delete the CRL, enter the **delete crl** command.
Step 7	(Optional) **show crypto ca certificates** **Example:** `switch(config-trustpoint)# show crypto ca certificates`	Displays the configured certificate chain and associated trustpoint.
Step 8	(Optional) **show crypto ca crl** *trustpoint* **Example:** `switch(config-trustpoint)# show crypto ca crl winca`	Displays the contents of the CRL list of the specified trustpoint.
Step 9	(Optional) **show user-account** **Example:** `switch(config-trustpoint)# show user-account`	Displays configured user account details.
Step 10	(Optional) **show users** **Example:** `switch(config-trustpoint)# show users`	Displays the users logged into the device.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config-trustpoint)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Legacy SSH Algorithm Support

You can configure support for legacy SSH security algorithms, message authentication codes (MACs), key types, and ciphers.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#?`
		Enters the global configuration mode.

Step 2	(Optional) **ssh kexalgos all** **Example:** `switch(config)# ssh kexalgos all`	Enables all supported KexAlgorithms which are the key exchange methods that are used to generate per-connection keys. Supported KexAlgorithmns are:

	Command or Action	Purpose
		• curve25519-sha256 • diffie-hellman-group-exchange-sha256 • diffie-hellman-group14-sha1 • diffie-hellman-group1-sha1 • ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521
Step 3	(Optional) **ssh macs all** **Example:** `switch(config)# ssh macs all`	Enables all supported MACs which are the message authentication codes used to detect traffic modification. Supported MACs are: • hmac-sha1 • hmac-sha2-256 • hmac-sha2-512
Step 4	(Optional) **ssh ciphers all** **Example:** `switch(config)# ssh ciphers all`	Enables all supported ciphers to encrypt the connection. Supported ciphers are: • aes128-cbc • aes192-cbc • aes256-cbc • aes128-ctr • aes192-ctr • aes256-ctr • aes256-gcm@openssh.com • aes128-gcm@openssh.com
Step 5	(Optional) **ssh keytypes all** **Example:** `switch(config)# ssh keytypes all`	Enables all supported PubkeyAcceptedKeyTypes which are the public key algorithms that the server can use to authenticate itself to the client. Supported key types are: • ecdsa-sha2-nistp256 • ecdsa-sha2-nistp384

	Command or Action	Purpose
		• ecdsa-sha2-nistp521 • ssh-dss • ssh-rsa

Changing the Default SSH Server Port

You can change the SSHv2 port number from the default port number 22. Encryptions used while changing the default SSH port provides you with connections that support stronger privacy and session integrity

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.

Step 3	**show sockets** *local-port-range*
	Example:
	`switch(config)# show sockets local port range (15001 - 58000)`
	`switch(config)# local port range (58001 - 63535) and nat port range (63536 - 65535)`
		Displays the available port range.

Step 4	**ssh port** *local-port* **Example:** `switch(config)# ssh port 58003`	Configures the port.
Step 5	**feature ssh** **Example:** `switch(config)# feature ssh`	Enables SSH.

Step 6	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 7	(Optional) **show running-config security all** **Example:**	Displays the security configuration.

	Command or Action	Purpose
	`switch# ssh port 58003`	
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Clearing SSH Hosts

When you download a file from a server using SCP or SFTP, or when you start an SSH session from this device to a remote host, you establish a trusted SSH relationship with that server. You can clear the list of trusted SSH servers for your user account.

Procedure

	Command or Action	Purpose
Step 1	**clear ssh hosts** **Example:** `switch# clear ssh hosts`	Clears the SSH host sessions and the known host file.

Disabling the SSH Server

By default, the SSH server is enabled on the Cisco NX-OS device. You can disable the SSH server to prevent SSH access to the switch.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show ssh server** **Example:**	Displays the SSH server configuration.

	Command or Action	Purpose
	`switch# show ssh server`	
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Deleting SSH Server Keys

You can delete SSH server keys on the Cisco NX-OS device after you disable the SSH server.

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

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.
Step 3	**no ssh key** [**dsa** | **rsa** | **ecdsa**] **Example:** `switch(config)# no ssh key rsa`	Deletes the SSH server key. The default is to delete all the SSH keys.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 5	(Optional) **show ssh key** **Example:** `switch# show ssh key`	Displays the SSH server key configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `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 NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	Displays user session information.
Step 2	**clear line** *vty-line* **Example:** `switch(config)# clear line pts/12`	Clears a user SSH session.

Configuring Telnet

This section describes how to configure Telnet on the Cisco NX-OS device.

Enabling the Telnet Server

You can enable the Telnet server on the Cisco NX-OS device. By default, the Telnet server is disabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature telnet** **Example:** `switch(config)# feature telnet`	Enables the Telnet server. The default is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show telnet server** **Example:** `switch# show telnet server`	Displays the Telnet server configuration.

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

Starting Telnet Sessions to Remote Devices

You can start Telnet sessions to connect to remote devices from the Cisco NX-OS device. You can start Telnet sessions using either IPv4 or IPv6.

Before you begin

Obtain the hostname or IP address for the remote device and, if needed, the username on the remote device.

Enable the Telnet server on the Cisco NX-OS device.

Enable the Telnet server on the remote device.

Procedure

	Command or Action	Purpose
Step 1	**telnet** {*ipv4-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*] **Example:** `switch# telnet 10.10.1.1`	Starts a Telnet session to a remote device using IPv4. The default port number is 23. The range is from 1 to 65535. The default VRF is the default VRF.
Step 2	**telnet6** {*ipv6-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*] **Example:** `switch# telnet6 2001:0DB8::ABCD:1 vrf management`	Starts a Telnet session to a remote device using IPv6. The default port number is 23. The range is from 1 to 65535. The default VRF is the default VRF.

Clearing Telnet Sessions

You can clear Telnet sessions from the Cisco NX-OS device.

Before you begin

Enable the Telnet server on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	Displays user session information.

	Command or Action	Purpose
Step 2	**clear line** *vty-line* **Example:** `switch(config)# clear line pts/12`	Clears a user Telnet session.

Verifying the SSH and Telnet Configuration

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

Command	Purpose
show ssh key [**dsa** | **rsa**] [**md5**]	Displays the SSH server keys.
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.
show ssh server	Displays the SSH server configuration.
show telnet server	Displays the Telnet server configuration.
show username *username* **keypair**	Displays the public key for the specified user.
show user-account	Displays configured user account details.
show users	Displays the users logged into the device.
show crypto ca certificates	Displays the configured CA certificate and associated trustpoint for X.509v3 certificate-based SSH authentication.
show crypto ca crl *trustpoint*	Displays the contents of the CRL list of the specified trustpoint for X.509v3 certificate-based SSH authentication.

Configuration Example for SSH

The following example shows how to configure SSH with an OpenSSH key:

Procedure

Step 1 Disable the SSH server.

Example:

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

Step 2 Generate an SSH server key.

Example:

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

Step 3 Enable the SSH server.

Example:

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

Step 4 Display the SSH server key.

Example:

Step 5 Specify the SSH public key in OpenSSH format.

Example:

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

Step 6 Save the configuration.

Example:

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

Configuration Example for SSH Passwordless File Copy

The following example shows how to copy files from a Cisco NX-OS device to a secure copy (SCP) or secure FTP (SFTP) server without a password:

Procedure

Step 1 Generate the SSH public and private keys and store them in the home directory of the Cisco NX-OS device for the specified user.

Example:

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

Step 2 Display the public key for the specified user.

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

Step 3 Export the public and private keys from the home directory of the Cisco NX-OS device to the specified bootflash directory.

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
.
.
```

Step 4 After copying these two files to another Cisco NX-OS device using the **copy scp** or **copy sftp** command, import them to the home directory of the Cisco NX-OS device.

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)#
```

Step 5 On the SCP or SFTP server, append the public key stored in key_rsa.pub to the authorized_keys file.

Example:

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

You can now copy files from the Cisco NX-OS device to the server without a password using standard SSH and SCP commands.

Step 6 (Optional) Repeat this procedure for the DSA keys.

Configuration Example for X.509v3 Certificate-Based SSH Authentication

The following example shows how to configure SSH authentication using X.509v3 certificates:

```
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
```

Additional References for SSH and Telnet

This section describes additional information related to implementing SSH and Telnet.

Related Documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus 3400-S NX-OS Unicast Routing Configuration Guide*