# Configuring Secure Shell

The Secure Shell (SSH) feature is an application and a protocol that provides a secure replacement to the Berkeley r-tools. The protocol secures sessions using standard cryptographic mechanisms, and the application can be used similarly to the Berkeley rexec and rsh tools. Two versions of SSH are available: SSH Version 1 and SSH Version 2. Unless otherwise noted, the term "SSH" denotes "SSH Version 1" only. For information about SSH Version 2, see the " Secure Shell Version 2 Support" feature module.


- Prerequisites for Configuring Secure Shell, on page 1
- Restrictions for Configuring Secure Shell, on page 2
- Information About Configuring Secure Shell , on page 3
- How to Configure Secure Shell, on page 5
- Configuration Examples for Secure Shell, on page 7
- Additional References for Secure Shell, on page 9
- Feature History for Configuring Secure Shell, on page 9


# Prerequisites for Configuring Secure Shell

**Note** Unless otherwise noted, the term "SSH" denotes "SSH Version 1" only.

- For SSH to work, the switch needs an Rivest, Shamir, and Adleman (RSA) public/private key pair. This is the same with Secure Copy Protocol (SCP), which relies on SSH for its secure transport.
- Download the required image on the device. The Secure Shell (SSH) server requires an IPsec (Data Encryption Standard [DES] or 3DES) encryption software image; the SSH client requires an IPsec (DES or 3DES) encryption software image.)
- Configure a hostname and host domain for your device by using the **hostname** and **ip domain name** commands in global configuration mode.
- Generate a Rivest, Shamir, and Adleman (RSA) key pair for your device. This key pair automatically enables SSH and remote authentication when the **crypto key generate rsa** command is entered in global configuration mode.

**Note** To delete the RSA key pair, use the **crypto key zeroize rsa** global configuration command. Once you delete the RSA key pair, you automatically disable the SSH server.

- Configure user authentication for local or remote access. You can configure authentication with or without authentication, authorization, and accounting (AAA).
- The Secure Shell (SSH) server requires an IPsec (Data Encryption Standard [DES] or 3DES) encryption software image; the SSH client requires an IPsec (DES or 3DES) encryption software image.)
- For Strict-KEX to function, devices must be running Cisco IOS-XE with SSH support enabled.
- Both client and server devices should be compatible with the strict-kex algorithms for Strict-KEX to be effective.

# Restrictions for Configuring Secure Shell

**Note** Unless otherwise noted, the term "SSH" denotes "SSH Version 1" only.

- The Secure Shell (SSH) server and SSH client are supported on Data Encryption Standard (DES) (56-bit) and 3DES (168-bit) data encryption software images only. In DES software images, DES is the only encryption algorithm available. In 3DES software images, both DES and 3DES encryption algorithms are available.
- Execution shell is the only application supported.
- The login banner is not supported in Secure Shell Version 1. It is supported in Secure Shell Version 2.
- The SFTP server is not supported.
- Starting from Cisco IOS XE Release 17.10, the following Key Exchange and MAC algorithms are removed from the default list:

  Key Exchange algorithm:

  - diffie-hellman-group14-sha1

  MAC algorithms:

  - hmac-sha1
  - hmac-sha2-256
  - hmac-sha2-512

**Note** You can use the **ip ssh server algorithm kex** command to configure the Key Exchange algorithm and the **ip ssh server algorithm mac** command to configure the MAC algorithms.

# Information About Configuring Secure Shell

Secure Shell (SSH) is a protocol that provides a secure, remote connection to a device. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated. This software release supports SSH Version 2 (SSHv2).

## SSH Server

**Note** Unless otherwise noted, the term "SSH" denotes "SSH Version 1" only.

The Secure Shell (SSH) Server feature enables an SSH client to make a secure, encrypted connection to a Cisco device. This connection provides functionality that is similar to that of an inbound Telnet connection. Before SSH, security was limited to Telnet security. SSH allows a strong encryption to be used with the Cisco software authentication. The SSH server in Cisco software works with publicly and commercially available SSH clients.

## SSH Integrated Client

**Note** Unless otherwise noted, the term "SSH" denotes "SSH Version 1" only.

The Secure Shell (SSH) Integrated Client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco device to make a secure, encrypted connection to another Cisco device or to any other device running the SSH server. This connection provides functionality similar to that of an outbound Telnet connection except that the connection is encrypted. With authentication and encryption, the SSH client allows for secure communication over an unsecured network.

The SSH client in Cisco software works with publicly and commercially available SSH servers. The SSH client supports the ciphers of Data Encryption Standard (DES), 3DES, and password authentication. User authentication is performed like that in the Telnet session to the device. The user authentication mechanisms supported for SSH are RADIUS, TACACS+, and the use of locally stored usernames and passwords.

**Note** The SSH client functionality is available only when the SSH server is enabled.

## RSA Authentication Support

Rivest, Shamir, and Adleman (RSA) authentication available in Secure Shell (SSH) clients is not supported on the SSH server for Cisco software by default. For more information about RSA authentication support, see the "Configuring a Device for SSH Version 2 Using RSA Pairs" section of the "Secure Shell Version 2 Support" module.

# SSH Servers, Integrated Clients, and Supported Versions

The Secure Shell (SSH) Integrated Client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco device to make a secure, encrypted connection to another Cisco device or to any other device running the SSH server. This connection provides functionality similar to that of an outbound Telnet connection except that the connection is encrypted. With authentication and encryption, the SSH client allows for secure communication over an unsecured network.

The SSH server and SSH integrated client are applications that run on the switch. The SSH server works with the SSH client supported in this release and with non-Cisco SSH clients. The SSH client works with publicly and commercially available SSH servers. The SSH client supports the ciphers of Data Encryption Standard (DES), 3DES, and password authentication.

**Note** The SSH client functionality is available only when the SSH server is enabled.

User authentication is performed like that in the Telnet session to the device. SSH also supports the following user authentication methods:

- TACACS+
- RADIUS
- Local authentication and authorization

# SSH Configuration Guidelines

Follow these guidelines when configuring the switch as an SSH server or SSH client:

- An RSA key pair generated by a SSHv1 server can be used by an SSHv2 server, and the reverse.
- If the SSH server is running on an active switch and the active switch fails, the new active switch uses the RSA key pair generated by the previous active switch.
- If you get CLI error messages after entering the **crypto key generate rsa** global configuration command, an RSA key pair has not been generated. Reconfigure the hostname and domain, and then enter the **crypto key generate rsa** command.
- When generating the RSA key pair, the message No host name specified might appear. If it does, you must configure a hostname by using the **hostname** command in global configuration mode.
- When generating the RSA key pair, the message No domain specified might appear. If it does, you must configure an IP domain name by using the **ip domain name** command in global configuration mode.
- When configuring the local authentication and authorization authentication method, make sure that AAA is disabled on the console.

# Strict-KEX

Strict-KEX is a protocol extension that strengthens security by ensuring that unexpected or out-of-sequence packets result in immediate connection termination and by resetting sequence numbers after key exchanges. To provide continuous protection, strict-KEX is enabled by default and cannot be disabled.

# How to Configure Secure Shell

## Setting Up the Device to Run SSH

Follow the procedure given below to set up your device to run SSH:

### Before you begin

Configure user authentication for local or remote access. This step is required.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **hostname** *hostname*<br>**Example:**<br>`Device(config)# hostname your_hostname` | Configures a hostname and IP domain name for your device.<br>**Note**<br>Follow this procedure only if you are configuring the device as an SSH server. |
| **Step 4** | **ip domain name** *domain_name*<br>**Example:**<br>`Device(config)# ip domain name your_domain` | Configures a host domain for your device. |
| **Step 5** | **crypto key generate rsa**<br>**Example:**<br>`Device(config)# crypto key generate rsa` | Enables the SSH server for local and remote authentication on the device and generates an RSA key pair. Generating an RSA key pair for the device automatically enables SSH.<br>We recommend that a minimum modulus size of 1024 bits.<br>When you generate RSA keys, you are prompted to enter a modulus length. A longer modulus length might be more secure, but it takes longer to generate and to use.<br>**Note** |

| | Command or Action | Purpose |
|---|---|---|
| | | Follow this procedure only if you are configuring the device as an SSH server. |
| **Step 6** | **exit**<br>**Example:**<br>`Device(config)# exit` | Exits global configuration mode and returns to privileged EXEC mode. |
| **Step 7** | **show ip ssh**<br>**Example:**<br>`Device# show ip ssh` | (Optional) Verifies that the SSH server is enabled and displays the version and configuration data for the SSH connection. |

# Configuring an SSH Server

**Note** Unless otherwise noted, the term "SSH" denotes "SSH Version 1" only.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **ip ssh** {**time-out** *seconds* \| **authentication-retries** *integer*}<br>**Example:**<br>`Device(config)# ip ssh time-out 30` | Configures Secure Shell (SSH) control parameters.<br>**Note**<br>This command can also be used to establish the number of password prompts provided to the user. The number is the lower of the following two values:<br>• Value proposed by the client using the **ssh -o numberofpasswordprompt** command.<br>• Value configured on the device using the **ip ssh authentication-retries** *integer*command, plus one. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **ip ssh rekey** {**time** *time* \| **volume** *volume*}<br>**Example:**<br>`Device(config)# ip ssh rekey time 108` | (Optional) Configures a time-based rekey or a volume-based rekey for SSH. |
| **Step 5** | **exit**<br>**Example:**<br>`Device(config)# exit` | Exits global configuration mode and returns to privileged EXEC mode. |
| **Step 6** | **show ip ssh**<br>**Example:**<br>`Device# show ip ssh` | (Optional) Verifies that the SSH server is enabled and displays the version and configuration data for the SSH connection. |

# Invoking an SSH Client

**Note** Unless otherwise noted, the term "SSH" denotes "SSH Version 1" only.

Perform this task to invoke the Secure Shell (SSH) client. The SSH client runs in user EXEC mode and has no specific configuration tasks.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **ssh -l** *username* **-vrf** *vrf-name* *ip-address*<br>**Example:**<br>`Device# ssh -l user1 -vrf vrf1 192.0.2.1` | Invokes the SSH client to connect to an IP host or address in the specified virtual routing and forwarding (VRF) instance. |

# Configuration Examples for Secure Shell

## Example: Configuring an SSH Server

**Note** Unless otherwise noted, the term "SSH" denotes "SSH Version 1" only.

The following is an example of the Secure Shell (SSH) control parameters configured for the server. In this example, the timeout interval of 30 seconds has been specified. This timeout interval is used during the SSH negotiation phase.

```
Device> enable
Device# configure terminal
Device(config)# ip ssh timeout 30
Device(config)# end
```

# Example: Invoking an SSH Client

**Note** Unless otherwise noted, the term “SSH” denotes “SSH Version 1” only.

In the following example, the Secure Shell (SSH) client has been invoked to connect to IP address 192.0.2.1 in the specified virtual routing and forwarding (VRF) instance:

```
Device> enable
Device# ssh -l user1 -vrf vrf1 192.0.2.1
```

# Example: Verifying SSH

**Note** Unless otherwise noted, the term “SSH” denotes “SSH Version 1” only.

To verify that the Secure Shell (SSH) server is enabled and to display the version and configuration data for your SSH connection, use the **show ip ssh** command. The following example shows that SSH is enabled:

```
Device# show ip ssh

SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
```

The following example shows that SSH is disabled:

```
Device# show ip ssh

%SSH has not been enabled
```

To verify the status of your SSH server connections, use the **show ssh** command. The following example shows the SSH server connections on the device when SSH is enabled:

```
Device# show ssh

Connection      Version     Encryption State Username
 0 1.5 3DES Session Started  guest
```

The following example shows that SSH is disabled:

```
Device# show ssh

%No SSH server connections running.
```

# Additional References for Secure Shell

### Related Documents

| Related Topic | Document Title |
|---|---|
| SSH Version 2 | Secure Shell Version 2 Support module in the *Security Configuration Guide* |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.<br>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.<br>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. | http://www.cisco.com/techsupport |

# Feature History for Configuring Secure Shell

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

| Release | Feature | Feature Information |
|---|---|---|
| Cisco IOS XE Everest 16.5.1a | Secure Shell | SSH is a protocol that provides a secure, remote connection to a device. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated..<br>Support for this feature was introduced on the C9500-12Q, C9500-16X, C9500-24Q, and C9500-40X models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Fuji 16.8.1a | Secure Shell | Support for this feature was introduced on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches. |
| Cisco IOS XE Cupertino 17.7.1 | Secure Shell | This feature was implemented on the C9500X-28C8D model. |

| Release | Feature | Feature Information |
|---|---|---|
| Cisco IOS XE Dublin 17.10.1b | Secure Shell | This feature was implemented on the C9500X-60L4D model. |
| Cisco IOS XE 17.17.x | Strict-KEX | Thisfeature was implemented on all models of the Cisco Catalyst 9500 Series Switches . |

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/.