



# Secure Shell Version 2 Support

---

**First Published: November 3, 2003**

**Last Updated: September 10, 2007**

The Secure Shell Version 2 Support feature allows you to configure Secure Shell (SSH) Version 2 (SSH Version 1 support was implemented in an earlier Cisco IOS software release). SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. Currently, the only reliable transport that is defined for SSH is TCP. SSH provides a means to securely access and securely execute commands on another computer over a network. The Secure Copy Protocol (SCP) feature that is provided with SSH also allows for the secure transfer of files.

## **Finding Feature Information in This Module**

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the “[Feature Information for Secure Shell Version 2 Support](#)” section on page 24.

## **Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images**

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

## Contents

- [Prerequisites for Secure Shell Version 2 Support, page 2](#)
- [Restrictions for Secure Shell Version 2 Support, page 2](#)
- [Information About Secure Shell Version 2 Support, page 2](#)
- [How to Configure Secure Shell Version 2 Support, page 3](#)
- [Configuration Examples for Secure Shell Version 2 Support, page 12](#)
- [Where to Go Next, page 14](#)
- [Additional References, page 14](#)
- [Command Reference, page 15](#)



---

**Americas Headquarters:**

**Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA**

© 2003–2007 Cisco Systems, Inc. All rights reserved.

## Prerequisites for Secure Shell Version 2 Support

Prior to configuring SSH, perform the following task:

- Download the required image on your router. The SSH server requires you to have a k9 (Triple Data Encryption Standard [3DES]) software image from Cisco IOS Release 12.3(4)T, 12.2(25)S, or 12.3(7)JA downloaded on your router.

**Note**

The SSH Version 2 server is supported in Cisco IOS Release 12.3(4)T, 12.3(2)XE, 12.2(25)S, and 12.3(7)JA; the SSH Version 2 client is supported beginning with Cisco IOS Release 12.3(7)T and is supported in Cisco IOS Release 12.3(7)JA. (The SSH client runs both the SSH Version 1 and Version 2 protocol and is supported in both k8 and k9 images in Cisco IOS Release 12.3(4)T.)

For more information on downloading a software image, refer to [Cisco IOS Configuration Fundamentals and Network Management Configuration Guide](#).

## Restrictions for Secure Shell Version 2 Support

- Rivest, Shamir, and Adelman (RSA) user authentication is not supported in the SSH server or SSH client for Cisco IOS software.
- SSH servers and SSH clients are supported in 3DES software images.
- Execution Shell, remote command execution, and Secure Copy Protocol (SCP) are the only applications supported.
- Compression is not supported.
- The RSA key-pair size must be greater than or equal to 768.

## Information About Secure Shell Version 2 Support

To configure SSH Version 2, you should understand the following concept:

- [Secure Shell Version 2, page 2](#)
- [SNMP Trap Generation, page 3](#)

## Secure Shell Version 2

The Secure Shell Version 2 Support feature allows you to configure SSH Version 2.

The configuration for the SSH Version 2 server is similar to the configuration for SSH Version 1. The command **ip ssh version** has been introduced so that you may define which version of SSH that you want to configure. If you do not configure this command, SSH by default runs in compatibility mode; that is, both SSH Version 1 and SSH Version 2 connections are honored.

**Note**

SSH Version 1 is a protocol that has never been defined in a standard. If you do not want your router to fall back to the undefined protocol (Version 1), you should use the **ip ssh version** command and specify Version 2.

The **ip ssh rsa keypair-name** command was also introduced in Cisco IOS Release 12.3(4)T so that you can enable a SSH connection using RSA keys that you have configured. Previously, SSH was linked to the first RSA keys that were generated (that is, SSH was enabled when the first RSA key pair was generated). The behavior still exists, but by using the **ip ssh rsa keypair-name** command, you can overcome that behavior. If you configure the **ip ssh rsa keypair-name** command with a key-pair name, SSH is enabled if the key pair exists, or SSH will be enabled if the key pair is generated later. If you use this command to enable SSH, you are not forced to configure a host name and a domain name, which was required in SSH Version 1 of the Cisco IOS software.

**Note**

The login banner is supported in Secure Shell Version 2, but it is not supported in Secure Shell Version 1.

## SNMP Trap Generation

Effective with Cisco IOS Release 12.4(17), Simple Network Management Protocol (SNMP) traps will be generated automatically when an SSH session terminates if the traps have been enabled and SNMP debugging has been turned on. For information about enabling SNMP traps, see the chapter “Configuring SNMP Support” in the *Cisco IOS Network Management Configuration Guide*.

**Note**

When configuring the **snmp-server host** command, the IP address must be the address of the PC that has the SSH (telnet) client and that has IP connectivity to the SSH server. For an example of an SNMP trap generation configuration, see the section “[Setting an SNMP Trap: Example](#).”

You must also turn on SNMP debugging using the **debug snmp packet** command to display the traps. The trap information includes information such as the number of bytes sent and the protocol that was used for the SSH session. For an example of SNMP debugging, see the section “[SNMP Debugging: Example](#).”

## How to Configure Secure Shell Version 2 Support

This section contains the following procedures:

- [Configuring a Router for SSH Version 2 Using a Host Name and Domain Name, page 4](#) (required)
- [Configuring a Router for SSH Version 2 Using RSA Key Pairs, page 5](#) (optional)
- [Starting an Encrypted Session with a Remote Device, page 6](#) (optional)
- [Verifying the Status of the Secure Shell Connection Using the show ssh Command, page 7](#) (optional)
- [Verifying the Secure Shell Status Using the show ip ssh Command, page 8](#) (optional)
- [Monitoring and Maintaining Secure Shell Version 2, page 9](#) (optional)

## Configuring a Router for SSH Version 2 Using a Host Name and Domain Name

To configure your router for SSH Version 2 using a host name and domain name, perform the following steps. You may also configure SSH Version 2 by using the RSA key pair configuration (See the section “[Configuring a Router for SSH Version 2 Using RSA Key Pairs](#)”).

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *hostname*
4. **ip domain-name** *name*
5. **crypto key generate rsa**
6. **ip ssh** [*timeout seconds* | *authentication-retries integer*]
7. **ip ssh version** [1 | 2]

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>hostname</b> <i>hostname</i>  <b>Example:</b> Router (config)# hostname cisco 7200	Configures a host name for your router.
Step 4	<b>ip domain-name</b> <i>name</i>  <b>Example:</b> Router (config)# ip domain-name cisco.com	Configures a domain name for your router.
Step 5	<b>crypto key generate rsa</b>  <b>Example:</b> Router (config)# crypto key generate rsa	Enables the SSH server for local and remote authentication.

	Command or Action	Purpose
Step 6	<pre>ip ssh [timeout seconds   authentication-retries integer]</pre> <p><b>Example:</b> Router (config)# ip ssh timeout 120</p>	(Optional) Configures SSH control variables on your router.
Step 7	<pre>ip ssh version [1   2]</pre> <p><b>Example:</b> Router (config)# ip ssh version 1</p>	(Optional) Specifies the version of SSH to be run on your router.

## Configuring a Router for SSH Version 2 Using RSA Key Pairs

To enable SSH Version 2 without configuring a host name or domain name, perform the following steps. SSH Version 2 will be enabled if the key pair that you configure already exists or if it is generated later. You may also configure SSH Version 2 by using the host name and domain name configuration (See the section “[Configuring a Router for SSH Version 2 Using a Host Name and Domain Name](#)”).

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh rsa keypair-name** *keypair-name*
4. **crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size*
5. **ip ssh [timeout seconds | authentication-retries integer]**
6. **ip ssh version [1 | 2]**

### DETAILED STEPS

Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable</p>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# configure terminal</p>	Enters global configuration mode.
Step 3	<pre>ip ssh rsa keypair-name keypair-name</pre> <p><b>Example:</b> Router (config)# ip ssh rsa keypair-name sshkeys</p>	Specifies which RSA keypair to use for SSH usage. <p><b>Note</b> A Cisco IOS router can have many RSA key pairs.</p>

Step 4	<pre>crypto key generate rsa usage-keys label key-label modulus modulus-size</pre> <p><b>Example:</b> Router (config)# crypto key generate rsa usage-keys label sshkeys modulus 768</p>	<p>Enables the SSH server for local and remote authentication on the router.</p> <p>For SSH Version 2, the modulus size must be at least 768 bits.</p> <p><b>Note</b> To delete the RSA key-pair, use the <b>crypto key zeroize rsa</b> command. After you have deleted the RSA command, you automatically disable the SSH server.</p>
Step 5	<pre>ip ssh [timeout seconds   authentication-retries integer]</pre> <p><b>Example:</b> Router (config)# ip ssh timeout 120</p>	<p>Configures SSH control variables on your router.</p>
Step 6	<pre>ip ssh version [1   2]</pre> <p><b>Example:</b> Router (config)# ip ssh version 1</p>	<p>Specifies the version of SSH to be run on a router.</p>

## Starting an Encrypted Session with a Remote Device

To start an encrypted session with a remote networking device, perform the following step. (You do not have to enable your router. SSH can be run in disabled mode.)



### Note

The device you wish to connect with must support a SSH server that has an encryption algorithm that is supported in Cisco IOS software.

## SUMMARY STEPS

1. `ssh [-v {1 | 2}] [-c {3des | aes128-cbc | aes192-cbc | aes256-cbc}] [-m {hmac-md5 | hmac-md5-96 | hmac-sha1 | hmac-sha1-96}] [I userid] [-o numberofpasswordprompts n] [-p port-num] {ip-addr | hostname} [command]`

## DETAILED STEPS

<p><b>Step 1</b></p> <pre>ssh [-v {1   2}] [-c {3des   aes128-cbc   aes192-cbc   aes256-cbc}] [-m {hmac-md5   hmac-md5-96   hmac-sha1   hmac-sha1-96}] [-o numberofpasswordprompts n] [-p port-num] {ip-addr   hostname} [command]</pre> <p><b>Example:</b> Router# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 -l user2 10.76.82.24</p> <p>Or</p> <p>The above example adheres to the SSH Version 2 conventions. A more natural and common way to start a session is by linking the username with the hostname. For example, the following configuration example provides an end result that is identical to that of the above example:</p> <pre>Router# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 user2@10.76.82.24</pre>	<p>Starts an encrypted session with a remote networking device.</p>
---	---

## Troubleshooting Tips

The **ip ssh version** command can be used for troubleshooting your SSH configuration. By changing versions, you can determine which SSH version has a problem.

## Verifying the Status of the Secure Shell Connection Using the show ssh Command

To display the status of the SSH connection on your router, use the **show ssh** command.

### SUMMARY STEPS

1. enable
2. show ssh

### DETAILED STEPS

<p><b>Step 1</b></p> <pre>enable</pre> <p><b>Example:</b> Router&gt; enable</p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<p><b>Step 2</b></p> <pre>show ssh</pre> <p><b>Example:</b> Router# show ssh</p>	<p>Displays the status of SSH server connections.</p>

## Examples

The following output examples from the **show ssh** command display status about various SSH Version 1 and Version 2 connections.

### Version 1 and Version 2 Connections

```
Router# show ssh
```

```
-----
Connection      Version Encryption      State                Username
 0              1.5      3DES              Session started     lab
Connection Version Mode Encryption Hmac                State
Username
1              2.0      IN aes128-cbc hmac-md5      Session started     lab
1              2.0      OUT aes128-cbc hmac-md5      Session started     lab
-----
```

### Version 2 Connection with No Version 1

```
Router# show ssh
```

```
-----
Connection Version Mode Encryption Hmac                State
Username
1              2.0      IN aes128-cbc hmac-md5      Session started     lab
1              2.0      OUT aes128-cbc hmac-md5      Session started     lab
%No SSHv1 server connections running.
-----
```

### Version 1 Connection with No Version 2

```
Router# show ssh
```

```
-----
Connection      Version Encryption      State                Username
 0              1.5      3DES              Session started     lab
%No SSHv2 server connections running.
-----
```

## Verifying the Secure Shell Status Using the show ip ssh Command

To verify your SSH configuration, perform the following steps.

### SUMMARY STEPS

1. enable
2. show ip ssh

## DETAILED STEPS

Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; enable </p>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<pre>show ip ssh</pre> <p><b>Example:</b> Router# show ip ssh </p>	<p>Displays the version and configuration data for SSH.</p>

## Examples

The following examples from the **show ip ssh** command display the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries.

### Version 1 and Version 2 Connections

```
-----
router# show ip ssh

3d06h: %SYS-5-CONFIG_I: Configured from console by consoleh
SSH Enabled - version 1.99
Authentication timeout: 120 secs; Authentication retries: 3
-----
```

### Version 2 Connection with No Version 1

```
-----
Router# show ip ssh

SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
-----
```

### Version 1 Connection with No Version 2

```
-----
Router# show ip ssh

3d06h: %SYS-5-CONFIG_I: Configured from console by console
SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
-----
```

## Monitoring and Maintaining Secure Shell Version 2

To display debug messages about the SSH connections, use the **debug ip ssh** command.

### SUMMARY STEPS

1. enable
2. debug ip ssh
3. debug snmp packet

## DETAILED STEPS

Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
Step 2	<b>debug ip ssh</b>  <b>Example:</b> Router# debug ip ssh	Displays debugging messages for SSH.
Step 3	<b>debug snmp packet</b>  <b>Example:</b> Router# debug snmp packet	Displays information about every SNMP packet sent or received by the router.

## Example

The following output from the **debug ip ssh** command shows that the digit 2 keyword has been assigned, signifying that it is an SSH Version 2 connection.

```
Router# debug ip ssh

00:33:55: SSH1: starting SSH control process
00:33:55: SSH1: sent protocol version id SSH-1.99-Cisco-1.25
00:33:55: SSH1: protocol version id is - SSH-2.0-OpenSSH_2.5.2p2
00:33:55: SSH2 1: send: len 280 (includes padlen 4)
00:33:55: SSH2 1: SSH2_MSG_KEXINIT sent
00:33:55: SSH2 1: ssh_receive: 536 bytes received
00:33:55: SSH2 1: input: packet len 632
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: ssh_receive: 96 bytes received
00:33:55: SSH2 1: partial packet 8, need 624, maclen 0
00:33:55: SSH2 1: input: padlen 11
00:33:55: SSH2 1: received packet type 20
00:33:55: SSH2 1: SSH2_MSG_KEXINIT received
00:33:55: SSH2: kex: client->server aes128-cbc hmac-md5 none
00:33:55: SSH2: kex: server->client aes128-cbc hmac-md5 none
00:33:55: SSH2 1: expecting SSH2_MSG_KEXDH_INIT
00:33:55: SSH2 1: ssh_receive: 144 bytes received
00:33:55: SSH2 1: input: packet len 144
00:33:55: SSH2 1: partial packet 8, need 136, maclen 0
00:33:55: SSH2 1: input: padlen 5
00:33:55: SSH2 1: received packet type 30
00:33:55: SSH2 1: SSH2_MSG_KEXDH_INIT received
00:33:55: SSH2 1: signature length 111
00:33:55: SSH2 1: send: len 384 (includes padlen 7)
00:33:55: SSH2: kex_derive_keys complete
00:33:55: SSH2 1: send: len 16 (includes padlen 10)
00:33:55: SSH2 1: newkeys: mode 1
00:33:55: SSH2 1: SSH2_MSG_NEWKEYS sent
00:33:55: SSH2 1: waiting for SSH2_MSG_NEWKEYS
00:33:55: SSH2 1: ssh_receive: 16 bytes received
00:33:55: SSH2 1: input: packet len 16
00:33:55: SSH2 1: partial packet 8, need 8, maclen 0
00:33:55: SSH2 1: input: padlen 10
00:33:55: SSH2 1: newkeys: mode 0
```

```
00:33:55: SSH2 1: received packet type 2100:33:55: SSH2 1: SSH2_MSG_NEWKEYS received
00:33:56: SSH2 1: ssh_receive: 48 bytes received
00:33:56: SSH2 1: input: packet len 32
00:33:56: SSH2 1: partial packet 16, need 16, maclen 16
00:33:56: SSH2 1: MAC #3 ok
00:33:56: SSH2 1: input: padlen 10
00:33:56: SSH2 1: received packet type 5
00:33:56: SSH2 1: send: len 32 (includes padlen 10)
00:33:56: SSH2 1: done calc MAC out #3
00:33:56: SSH2 1: ssh_receive: 64 bytes received
00:33:56: SSH2 1: input: packet len 48
00:33:56: SSH2 1: partial packet 16, need 32, maclen 16
00:33:56: SSH2 1: MAC #4 ok
00:33:56: SSH2 1: input: padlen 9
00:33:56: SSH2 1: received packet type 50
00:33:56: SSH2 1: send: len 32 (includes padlen 13)
00:33:56: SSH2 1: done calc MAC out #4
00:34:04: SSH2 1: ssh_receive: 160 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #5 ok
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 50
00:34:04: SSH2 1: send: len 16 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #5
00:34:04: SSH2 1: authentication successful for lab
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #6 ok
00:34:04: SSH2 1: input: padlen 6
00:34:04: SSH2 1: received packet type 2
00:34:04: SSH2 1: ssh_receive: 64 bytes received
00:34:04: SSH2 1: input: packet len 48
00:34:04: SSH2 1: partial packet 16, need 32, maclen 16
00:34:04: SSH2 1: MAC #7 ok
00:34:04: SSH2 1: input: padlen 19
00:34:04: SSH2 1: received packet type 90
00:34:04: SSH2 1: channel open request
00:34:04: SSH2 1: send: len 32 (includes padlen 10)
00:34:04: SSH2 1: done calc MAC out #6
00:34:04: SSH2 1: ssh_receive: 192 bytes received
00:34:04: SSH2 1: input: packet len 64
00:34:04: SSH2 1: partial packet 16, need 48, maclen 16
00:34:04: SSH2 1: MAC #8 ok
00:34:04: SSH2 1: input: padlen 13
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: pty-req request
00:34:04: SSH2 1: setting TTY - requested: height 24, width 80; set: height 24,
width 80
00:34:04: SSH2 1: input: packet len 96
00:34:04: SSH2 1: partial packet 16, need 80, maclen 16
00:34:04: SSH2 1: MAC #9 ok
00:34:04: SSH2 1: input: padlen 11
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: x11-req request
00:34:04: SSH2 1: ssh_receive: 48 bytes received
00:34:04: SSH2 1: input: packet len 32
00:34:04: SSH2 1: partial packet 16, need 16, maclen 16
00:34:04: SSH2 1: MAC #10 ok
00:34:04: SSH2 1: input: padlen 12
00:34:04: SSH2 1: received packet type 98
00:34:04: SSH2 1: shell request
00:34:04: SSH2 1: shell message received
00:34:04: SSH2 1: starting shell for vty
```

```

00:34:04: SSH2 1: send: len 48 (includes padlen 18)
00:34:04: SSH2 1: done calc MAC out #7
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #11 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #8
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #12 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #9
00:34:07: SSH2 1: ssh_receive: 48 bytes received
00:34:07: SSH2 1: input: packet len 32
00:34:07: SSH2 1: partial packet 16, need 16, maclen 16
00:34:07: SSH2 1: MAC #13 ok
00:34:07: SSH2 1: input: padlen 17
00:34:07: SSH2 1: received packet type 94
00:34:07: SSH2 1: send: len 32 (includes padlen 17)
00:34:07: SSH2 1: done calc MAC out #10
00:34:08: SSH2 1: ssh_receive: 48 bytes received
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #14 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 17)
00:34:08: SSH2 1: done calc MAC out #11
00:34:08: SSH2 1: ssh_receive: 48 bytes received
00:34:08: SSH2 1: input: packet len 32
00:34:08: SSH2 1: partial packet 16, need 16, maclen 16
00:34:08: SSH2 1: MAC #15 ok
00:34:08: SSH2 1: input: padlen 17
00:34:08: SSH2 1: received packet type 94
00:34:08: SSH2 1: send: len 32 (includes padlen 16)
00:34:08: SSH2 1: done calc MAC out #12
00:34:08: SSH2 1: send: len 48 (includes padlen 18)
00:34:08: SSH2 1: done calc MAC out #13
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
00:34:08: SSH2 1: done calc MAC out #14
00:34:08: SSH2 1: send: len 16 (includes padlen 6)
00:34:08: SSH2 1: done calc MAC out #15
00:34:08: SSH1: Session terminated normally

```

## Configuration Examples for Secure Shell Version 2 Support

This section provides the following configuration examples:

- [Configuring Secure Shell Version 1: Example, page 13](#)
- [Configuring Secure Shell Version 2: Example, page 13](#)
- [Configuring Secure Shell Versions 1 and 2: Example, page 13](#)
- [Starting an Encrypted Session with a Remote Device: Example, page 13](#)

- [Setting an SNMP Trap: Example, page 13](#)
- [SNMP Debugging: Example, page 13](#)

## Configuring Secure Shell Version 1: Example

```
Router# configure terminal
Router (config)# ip ssh version 1
c7200-25-2013(config)# end
```

## Configuring Secure Shell Version 2: Example

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip ssh version 2
Router(config)# end
```

## Configuring Secure Shell Versions 1 and 2: Example

```
Router# configure terminal
Router (config)# no ip ssh version
Router (config)# end
```

## Starting an Encrypted Session with a Remote Device: Example

```
Router# ssh -v 2 -c aes256-cbc -m hmac-sha1-160 -l shaship 10.76.82.24
```

## Setting an SNMP Trap: Example

The following shows that an SNMP trap has been set. The trap notification is generated automatically when the SSH session terminates. For an example of SNMP trap debug output, see the section “[SNMP Debugging: Example](#).”

```
Router# snmp-server
Router# snmp-server host a.b.c.d public tty
```

Where a.b.c.d is the IP address of the SSH client.

## SNMP Debugging: Example

The following is sample output using the **debug snmp packet** command. The output provides SNMP trap information for an SSH session.

```
router1# debug snmp packet

SNMP packet debugging is on

router1# ssh -l lab 10.0.0.2

Password:
```

```

router2# exit

[Connection to 10.0.0.2 closed by foreign host]
router1#
*Jul 18 10:18:42.619: SNMP: Queuing packet to 10.0.0.2
*Jul 18 10:18:42.619: SNMP: V1 Trap, ent cisco, addr 10.0.0.1, gentrap 6, spectrap 1
local.9.3.1.1.2.1 = 6
tcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 4
ltcpConnEntry.5.10.0.0.1.22.10.0.0.2.55246 = 1015
ltcpConnEntry.1.10.0.0.1.22.10.0.0.2.55246 = 1056
ltcpConnEntry.2.10.0.0.1.22.10.0.0.2.55246 = 1392
local.9.2.1.18.2 = lab
*Jul 18 10:18:42.879: SNMP: Packet sent via UDP to 10.0.0.2
router1#

```

## Where to Go Next

You have to use a SSH remote device that supports SSH Version 2, and you have to connect to a Cisco IOS router.

## Additional References

The following sections provide references related to Secure Shell Version 2.

## Related Documents

Related Topic	Document Title
AAA	<a href="#">“Authentication, Authorization, and Accounting (AAA)”</a> section of <i>Cisco IOS Security Configuration Guide</i>
Configuring a host name and host domain	<a href="#">“Configuring Secure Shell”</a> chapter in the <i>Cisco IOS Security Configuration Guide</i>
Configuring Secure Shell	<a href="#">“Configuring Secure Shell”</a> chapter of <i>Cisco IOS Security Configuration Guide</i>
Debugging commands	<i>Cisco IOS Debug Command Reference</i> , Release 12.4T
Downloading a Cisco software image	<i>Cisco IOS Configuration Fundamentals and Network Management Configuration Guide</i>
IOS configuration fundamentals	<i>Cisco IOS Configuration Fundamentals and Network Management Configuration Guide</i> and <i>Cisco IOS Configuration Fundamentals and Network Management Command Reference</i>
IPSec	<a href="#">“IP Security and Encryption”</a> section of <i>Cisco IOS Security Configuration Guide</i>
Security commands	<i>Cisco IOS Security Command Reference</i> , Release 12.4 T
SNMP, configuring traps	<a href="#">“Configuring SNMP Support”</a> chapter in <i>Cisco IOS Network Management Configuration Guide</i>

## Standards

Standards	Title
Internet Engineering Task Force (IETF) Secure Shell Version 2 Draft Standards	<a href="#">Internet Engineering Task Force website</a>

## MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

## RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

## Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>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.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a>

## Command Reference

This section documents only commands that are new or modified.

- [ip ssh rsa keypair-name](#)
- [ip ssh version](#)
- [ssh](#)

## ip ssh rsa keypair-name

To specify which Rivest, Shimar, and Adelman (RSA) key pair to use for a Secure Shell (SSH) connection, use the **ip ssh rsa keypair-name** command in global configuration mode. To disable the key pair that was configured, use the **no** form of this command.

**ip ssh rsa keypair-name** *keypair-name*

**no ip ssh rsa keypair-name** *keypair-name*

### Syntax Description

<i>keypair-name</i>	Name of the key pair.
---------------------	-----------------------

### Defaults

If this command is not configured, SSH will use the first RSA key pair that is enabled.

### Command Modes

Global configuration

### Command History

Release	Modification
12.3(4)T	This command was introduced.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.3(7)JA	This command was integrated into Cisco IOS Release 12.3(7)JA.
12.0(32)SY	This command was integrated into Cisco IOS Release 12.0(32)SY.

### Usage Guidelines

Using the **ip ssh rsa keypair-name** command, you can enable an SSH connection using RSA keys that you have configured using the *keypair-name* argument. Previously, SSH was tied to the first RSA keys that were generated (that is, SSH was enabled when the first RSA key pair was generated). The previous behavior still exists but by using the **ip ssh rsa keypair-name** command, you can overcome that behavior. If you configure the **ip ssh rsa keypair-name** command with a key pair name, SSH is enabled if the key pair exists, or SSH will be enabled if the key pair is generated later. If you use this command, you are not forced to configure a host name and a domain name.



#### Note

A Cisco IOS router can have many RSA key pairs.

### Examples

The following example shows that the **ip ssh rsa keypair-name** command has been used to specify the RSA key pair “sshkeys” for a SSH connection:

```
Router# configure terminal
Router (config)# ip ssh rsa keypair-name sshkeys
```

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>debug ip ssh</b>	Displays debug messages for SSH.
<b>disconnect ssh</b>	Terminates a SSH connection on your router.
<b>ip ssh</b>	Configures SSH control parameters on your router.
<b>ip ssh version</b>	Specifies the version of SSH to be run on a router.
<b>show ip ssh</b>	Displays the SSH connections of your router.

# ip ssh version

To specify the version of Secure Shell (SSH) to be run on a router, use the **ip ssh version** command in global configuration mode. To disable the version of SSH that was configured and to return to compatibility mode, use the **no** form of this command.

**ip ssh version [1 | 2]**

**no ip ssh version [1 | 2]**

## Syntax Description

<b>1</b>	(Optional) Router runs only SSH Version 1.
<b>2</b>	(Optional) Router runs only SSH Version 2.

## Defaults

If this command is not configured, SSH operates in compatibility mode, that is, Version 1 and Version 2 are both supported.

## Command Modes

Global configuration

## Command History

Release	Modification
12.3(4)T	This command was introduced.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.3(7)JA	This command was integrated into Cisco IOS Release 12.3(7)JA.
12.0(32)SY	This command was integrated into Cisco IOS Release 12.0(32)SY.

## Usage Guidelines

You can use this command with the **2** keyword to ensure that your router will not inadvertently establish a weaker SSH Version 1 connection.

## Examples

The following example shows that only SSH Version 1 support is configured:

```
Router (config)# ip ssh version 1
```

The following example shows that only SSH Version 2 is configured:

```
Router (config)# ip ssh version 2
```

The following example shows that SSH Versions 1 and 2 are configured:

```
Router (config)# no ip ssh version
```

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>debug ip ssh</b>	Displays debug messages for SSH.
<b>disconnect ssh</b>	Terminates a SSH connection on your router.
<b>ip ssh</b>	Configures SSH control parameters on your router.
<b>ip ssh rsa keypair-name</b>	Specifies which RSA key pair to use for a SSH connection.
<b>show ip ssh</b>	Displays the SSH connections of your router.

# ssh

To start an encrypted session with a remote networking device, use the **ssh** command in privileged EXEC or user EXEC mode.

```
ssh [-v {1 | 2}] [-c {3des | aes128-cbc | aes192-cbc | aes256-cbc}] [-l userid | -l userid:number
ip-address | -l userid:rotarynumber ip-address] [-m {hmac-md5 | hmac-md5-96 | hmac-sha1
| hmac-sha1-96}] [-o numberofpasswordprompts n] [-p port-num] [ip-addr | hostname]
[command]
```

## Syntax Description

<b>-v</b>	(Optional) Specifies the version of Secure Shell (SSH) to use to connect to the server. <ul style="list-style-type: none"> <li>• <b>1</b>—Connects using SSH Version 1.</li> <li>• <b>2</b>—Connects using SSH Version 2.</li> </ul>
<b>-c {3des   aes128-cbc   aes192-cbc   aes256-cbc}</b>	(Optional) Specifies the crypto algorithms Data Encryption Standard (DES), Triple DES (3DES), or Advanced Encryption Standard (AES) to use for encrypting data. AES algorithms supported are aes128-cbc, aes192-cbc, and aes256-cbc. <ul style="list-style-type: none"> <li>• To use SSH Version 1, you must have an encryption image running on the router. Cisco software images that include encryption have the designators “k8” (DES) or “k9” (3DES).</li> <li>• SSH Version 2 supports only the following crypto algorithms: aes128-cbc, aes192-cbc, aes256-cbc, and 3des-cbc. SSH Version 2 is supported only in 3DES images.</li> <li>• If you do not specify the <b>-c</b> keyword, during negotiation the remote networking device sends all the supported crypto algorithms.</li> <li>• If you configure the <b>-c</b> keyword and the server does not support the argument that you have shown (des, 3des, aes128-cbc, aes192-cbc, or aes256-cbc), the remote networking device closes the connection.</li> </ul>
<b>-l <i>userid</i></b>	(Optional) Specifies the user ID to use when logging in on the remote networking device running the SSH server. If no user ID is specified, the default is the current user ID.
<b>-l <i>userid:number ip-address</i></b>	(Optional) Specifies the user ID when configuring reverse SSH by including port information in the <i>userid</i> field. <ul style="list-style-type: none"> <li>• <b>:</b>—Signifies that a port number and terminal IP address will follow the user ID.</li> <li>• <i>number</i>—Terminal or auxiliary line number.</li> <li>• <i>ip-address</i>—IP address of the terminal server.</li> </ul> <p><b>Note</b> The <i>userid</i> argument and <i>:number ip-address</i> delimiter and arguments must be used if you are configuring reverse SSH by including port information in the <i>userid</i> field (a method that is easier than the longer method of listing each terminal or auxiliary line on a separate command configuration line).</p>

<b>-l</b> <i>userid:rotarynumber</i> <i>ip-address</i>	(Optional) Specifies that the terminal lines are to be grouped under the rotary group for reverse SSH. <ul style="list-style-type: none"> <li>• <b>:</b>—Signifies that a rotary group number and terminal IP address will follow.</li> <li>• <i>number</i>—Terminal or auxiliary line number.</li> <li>• <i>ip-address</i>—IP address of the terminal server.</li> </ul> <p><b>Note</b> The <i>userid</i> argument and <b>:rotary</b>{<i>number</i>} {<i>ip-address</i>} delimiter and arguments must be used if you are configuring reverse SSH by including rotary information in the <i>userid</i> field (a process that is easier than the longer process of listing each terminal or auxiliary line on a separate command configuration line).</p>
<b>-m</b> { <b>hmac-md5</b>   <b>hmac-md5-96</b>   <b>hmac-sha1</b>   <b>hmac-sha1-96</b> }	(Optional) Specifies a Hashed Message Authentication Code (HMAC) algorithm. <ul style="list-style-type: none"> <li>• SSH Version 1 does not support HMACs.</li> <li>• If you do not specify the <b>-m</b> keyword, the remote device sends all the supported HMAC algorithms during negotiation. If you specify the <b>-m</b> keyword and the server does not support the argument that you have shown (hmac-md5, hmac-md5-96, hmac-sha1, and hmac-sha1-96), the remote device closes the connection.</li> </ul>
<b>-o</b> <b>numberofpasswordprompts</b> <i>n</i>	(Optional) Specifies the number of password prompts that the software generates before ending the session. The SSH server may also apply a limit to the number of attempts. If the limit set by the server is less than the value specified by the <b>-o numberofpasswordprompts</b> keyword, the limit set by the server takes precedence. The default is 3 attempts, which is also the Cisco IOS SSH server default. The range of values is from 1 to 5.
<b>-p</b> <i>port-num</i>	(Optional) Indicates the desired port number for the remote host. The default port number is 22.
<i>ip-addr</i>   <i>hostname</i>	Specifies the IPv4 or IPv6 address or host name of the remote networking device.
<i>command</i>	(Optional) Specifies the Cisco IOS command that you want to run on the remote networking device. If the remote host is not running Cisco IOS software, this may be any command recognized by the remote host. If the command includes spaces, you must enclose the command in quotation marks.

**Command Default**

Disabled

**Command Modes**User EXEC  
Privileged EXEC

**Command History**

Release	Modification
12.1(3)T	This command was introduced.
12.2(8)T	Support for IPv6 addresses was added.
12.0(21)ST	IPv6 address support was integrated into Cisco IOS Release 12.0(21)ST.
12.0(22)S	IPv6 address support was integrated into Cisco IOS Release 12.0(22)S.
12.2(14)S	IPv6 address support was integrated into Cisco IOS Release 12.2(14)S.
12.2(17a)SX	This command was integrated into Cisco IOS Release 12.2(17a)SX.
12.3(7)T	This command was expanded to include Secure Shell Version 2 support. The <b>-c</b> keyword was expanded to include support for the following cryptic algorithms: aes128-cbc, aes192-cbc, and aes256-cbc. The <b>-m</b> keyword was added, with the following algorithms: hmac-md5, hmac-md5-96, hmac-sha1, and hmac-sha1-96. The <b>-v</b> keyword and arguments <b>1</b> and <b>2</b> were added.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.3(11)T	The <b>-l userid:number ip-address</b> and <b>-l userid:rotarynumber ip-address</b> keyword and argument options were added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(25)SG	This command was integrated into Cisco IOS Release 12.2(25)SG.
12.3(7)JA	This command was integrated into Cisco IOS Release 12.3(7)JA.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.0(32)SY	This command was integrated into Cisco IOS Release 12.0(32)SY.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

**Usage Guidelines**

The **ssh** command enables a Cisco router to make a secure, encrypted connection to another Cisco router or device running an SSH Version 1 or Version 2 server. This connection provides functionality that is similar to that of an outbound Telnet connection except that the connection is encrypted. With authentication and encryption, the SSH client allows for a secure communication over an insecure network.

**Note**

- SSH 1 is supported on 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.
- SSH Version 2 supports only the following crypto algorithms: aes128-cbc, aes192-cbc, and aes256-cbc. SSH Version 2 is supported only in 3DES images.
- SSH Version 1 does not support HMAC algorithms.

**Examples**

The following example illustrates the initiation of a secure session between the local router and the remote host HQhost to run the **show users** command. The result of the **show users** command is a list of valid users who are logged in to HQhost. The remote host will prompt for the adminHQ password to authenticate the user adminHQ. If the authentication step is successful, the remote host will return the result of the **show users** command to the local router and will then close the session.

```
ssh -l adminHQ HQhost "show users"
```

The following example illustrates the initiation of a secure session between the local router and the edge router HQedge to run the **show ip route** command. In this example, the edge router prompts for the adminHQ password to authenticate the user. If the authentication step is successful, the edge router will return the result of the **show ip route** command to the local router.

```
ssh -l adminHQ HQedge "show ip route"
```

The following example shows the SSH client using 3DES to initiate a secure remote command connection with the HQedge router. The SSH server running on HQedge authenticates the session for the admin7 user on the HQedge router using standard authentication methods. The HQedge router must have SSH enabled for authentication to work.

```
ssh -l admin7 -c 3des -o numberofpasswordprompts 5 HQedge
```

The following example shows a secure session between the local router and a remote IPv6 router with the address 3ffe:1111:2222:1044::72 to run the **show running-config** command. In this example, the remote IPv6 router prompts for the adminHQ password to authenticate the user. If the authentication step is successful, the remote IPv6 router will return the result of the **show running-config** command to the local router and will then close the session.

```
ssh -l adminHQ 3ffe:1111:2222:1044::72 "show running-config"
```



#### Note

A hostname that maps to the IPv6 address 3ffe:1111:2222:1044::72 could have been used in the last example.

The following example shows a SSH Version 2 session using the crypto algorithm aes256-cbc and an HMAC of hmac-sha1-96. The user ID is user2, and the IP address is 10.76.82.24.

```
ssh -v 2 -c aes256-cbc -m hmac-sha1-96 -l user2 10.76.82.24
```

The following example shows that reverse SSH has been configured on the SSH client:

```
ssh -l lab:1 router.example.com
```

The following command shows that Reverse SSH will connect to the first free line in the rotary group:

```
ssh -l lab:rotary1 router.example.com
```

#### Related Commands

Command	Description
<b>ip ssh</b>	Configures SSH server control parameters on the router.
<b>show ip ssh</b>	Displays the version and configuration data for SSH.
<b>show ssh</b>	Displays the status of SSH server connections.

# Feature Information for Secure Shell Version 2 Support

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.


**Note**

Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

**Table 1** Feature Information for Secure Shell Version 2 Support

Feature Name	Releases	Feature Information
Secure Shell Version 2 Support	12.3(4)T 12.2(25)S	The Secure Shell Version 2 Support feature allows you to configure Secure Shell (SSH) Version 2 (SSH Version 1 support was implemented in an earlier Cisco IOS software release). SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities.
Secure Shell Version 2 Client and Server Support	12.3(7)JA 12.0(32)SY	This feature was integrated into Cisco IOS Release 12.3(7)JA.
Secure Shell Version 2 Client and Server Support	12.4(17)	The Cisco IOS image was updated to provide for the automatic generation of SNMP traps when an SSH session terminates.  For information about this feature, see the following section: <ul style="list-style-type: none"> <li>• “SNMP Trap Generation” section on page 3</li> <li>• “SNMP Debugging: Example” section on page 13</li> </ul>

---

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0711R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2003–2007 Cisco Systems, Inc. All rights reserved.

