

CISCO Implementing Secure Socket Layer User Guide

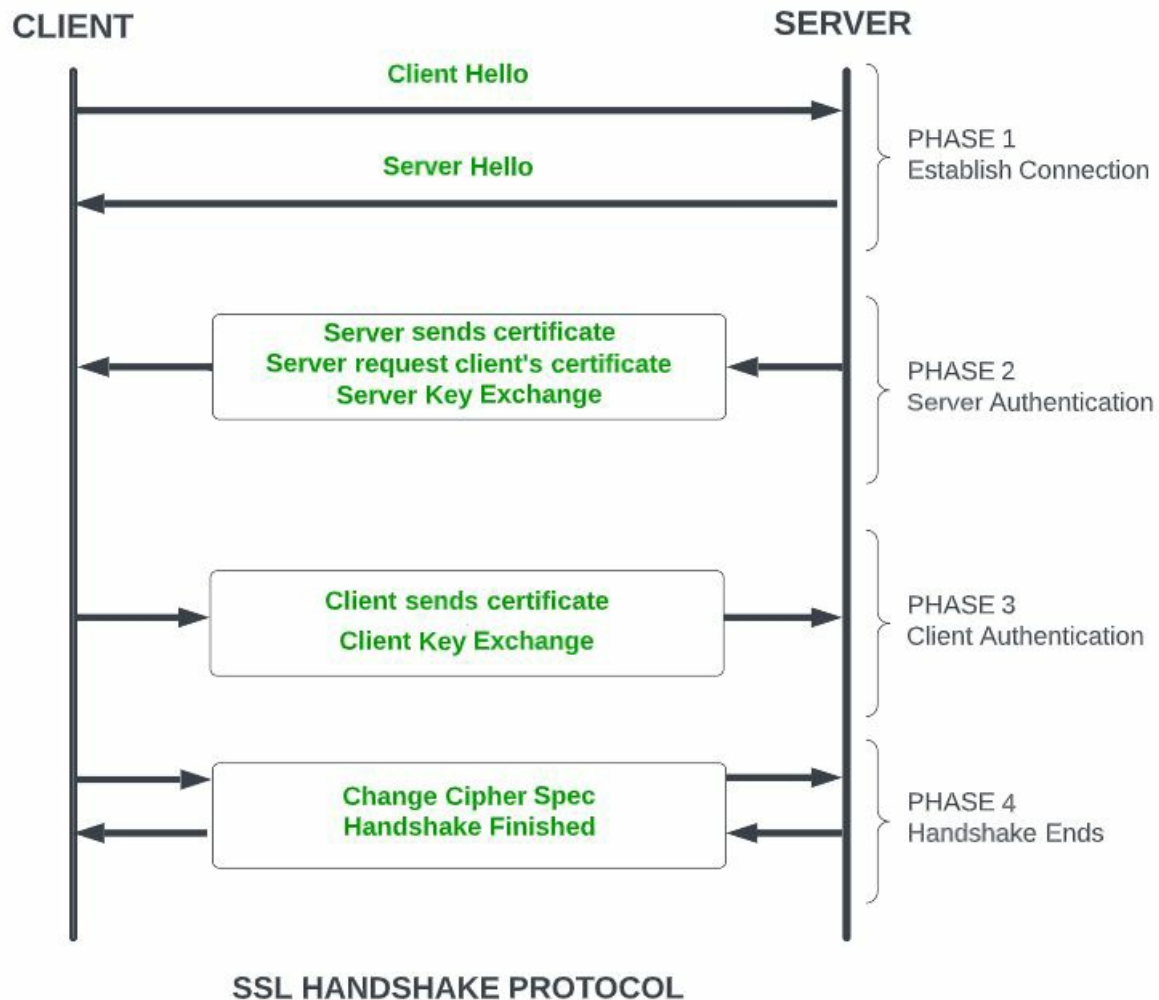
[Home](#) » [Cisco](#) » CISCO Implementing Secure Socket Layer User Guide 

Contents

- [1 CISCO Implementing Secure Socket Layer](#)
- [2 Product Information](#)
- [3 Specifications](#)
- [4 Prerequisites to Implementing Secure Shell](#)
- [5 SFTP Feature Overview](#)
- [6 SUMMARY STEPS](#)
- [7 Configuring the SSH Client](#)
- [8 Documents / Resources](#)
 - [8.1 References](#)
- [9 Related Posts](#)



CISCO Implementing Secure Socket Layer



Product Information

Specifications

- Product Name: Secure Socket Layer (SSL) module
- Protocol: SSL and Transport Layer Security (TLS)
- Authentication: Mutual authentication
- Integrity: Hash function
- Encryption: Public and private key cryptography
- Certificates: Issued by certification authorities (CAs) such as VeriSign or Thawte
- Public Key Infrastructure (PKI) support
- Advanced Encryption Standard (AES) encryption support
- Compatible with Cisco IOS XR Software
- Release History: Release 2.0, Release 3.8.0

Product Usage Instructions

Section 1: Implementing Secure Socket Layer

The Secure Socket Layer (SSL) protocol and Transport Layer Security (TLS) provide secure communication between a client and server. Follow the steps below to implement SSL:

1. Ensure you have the necessary prerequisites for implementing Secure Socket Layer.
2. Obtain certificates from certification authorities (CAs) like VeriSign or Thawte. Certificates include the authority's name, entity name, public key, and expiration date.
3. Generate public-private key pairs for encryption and decryption of information. The public key can be shared freely, but the private key should never be given out.
4. For detailed PKI commands, refer to the "Public Key Infrastructure Commands" module in the Cisco IOS XR System Security Command Reference for the Cisco CRS Router.
5. For SSL commands, refer to the "Secure Socket Layer Protocol Commands" module in the Cisco IOS XR System Security Command Reference for the Cisco CRS Router.

Prerequisites to Implementing Secure Shell

To implement Secure Shell (SSH), make sure you meet the following prerequisites:

- Ensure you have the necessary prerequisites for implementing Secure Socket Layer.

Information About Implementing Secure Shell

Secure Shell (SSH) enables secure, encrypted connections to a Cisco router. The following concepts are important to understand when implementing SSH:

- **SSH Server:** This enables an SSH client to make a secure, encrypted connection to a Cisco router. It provides similar functionality to an inbound Telnet connection but with stronger encryption. The SSH server works with publicly and commercially available SSH clients.
- **SSH Client:** Works with publicly and commercially available SSH servers. It supports ciphers such as AES, 3DES, MD5, SHA1, and password-based authentication.

FAQ

- **Q: What is a Secure Socket Layer (SSL)?**
 - A: Secure Socket Layer (SSL) is an application-level protocol that provides secure communication between a client and server. It allows for mutual authentication, integrity using hash functions, and encryption for privacy.
- **Q: What are certificates in SSL?**
 - A: Certificates are like digital ID cards that prove the identity of the server to clients. They are issued by certification authorities (CAs) and contain information such as the authority's name, entity name, public key, and expiration date.
- **Q: What are public and private keys in SSL?**
 - A: Public and private keys are ciphers used for encryption and decryption of information. The public key can be freely shared, while the private key should never be given out. Data encrypted with the public key can only be decrypted with the corresponding private key.
- **Q: What is Secure Shell (SSH)?**
 - A: Secure Shell (SSH) is a protocol that enables secure, encrypted connections to a Cisco router. It provides stronger security compared to Telnet and supports ciphers like AES, 3DES, MD5, SHA1, and password-based authentication.

This module describes how to implement SSL.

The Secure Socket Layer (SSL) protocol and Transport Layer Security (TLS) are application-level protocols that provide for secure communication between a client and server by allowing mutual authentication, the use of hash for integrity, and encryption for privacy. SSL and TLS rely on certificates, public keys, and private keys.

Certificates are similar to digital ID cards. They prove the identity of the server to clients. Certificates are issued by certification authorities (CAs), such as VeriSign or Thawte. Each certificate includes the name of the authority that issued it, the name of the entity to which the certificate was issued, the entity's public key, and time stamps that indicate the certificate's expiration date. Public and private keys are the ciphers used to encrypt and decrypt information. Although the public key is shared quite freely, the private key is never given out. Each public-private key pair works together: Data encrypted with the public key can be decrypted only with the private key.

Note

For a complete description of the Public Key Infrastructure (PKI) commands used here, see the Public Key Infrastructure Commands module in Cisco IOS XR System Security Command Reference for the Cisco CRS Router. For information on SSL commands, see the Secure Socket Layer Protocol Commands on the Cisco IOS XR Software Software module of Cisco IOS XR System Security Command Reference for the Cisco CRS Router.

Feature History for Implementing Secure Socket Layer

Release	Modification
Release 2.0	This feature was introduced.
Release 3.8.0	Advanced Encryption Standard (AES) encryption support was added on the SSL server.

Prerequisites to Implementing Secure Shell

The following prerequisites are required to implement Secure Shell:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Download the required image on your router. The SSH server and SSH client require you to have a crypto package (data encryption standard [DES], 3DES, and AES) from Cisco downloaded on your router.
- To run an SSHv2 server, you must have a VRF. This may be the default VRF or a specific VRF. VRF changes are applicable only to the SSH v2 server.
- Configure user authentication for local or remote access. You can configure authentication with or without authentication, authorization, and accounting (AAA). For more information, see the Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software module in the Cisco IOS XR System Security Command Reference for the Cisco CRS Router publication and Configuring AAA Services on Cisco IOS XR Software module in the Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router publication.
- AAA authentication and authorization must be configured correctly for Secure Shell File Transfer Protocol (SFTP) to work.

Restrictions for Implementing Secure Shell

The following are some basic SSH restrictions and limitations of the SFTP feature:

- A VRF is not accepted as an inband if that VRF is already set as an out-of-band VRF. SSH v1 continues to bind only to the default VRF.
- In order for an outside client to connect to the router, the router needs to have an RSA (for SSHv1 or SSHv2) or DSA (for SSHv2) key pair configured. DSA and RSA keys are not required if you are initiating an SSH client connection from the router to an outside routing device. The same is true for SFTP: DSA and RSA keys are not required because SFTP operates only in client mode.
- In order for SFTP to work properly, the remote SSH server must enable the SFTP server functionality. For example, the SSHv2 server is configured to handle the SFTP subsystem with a line such as
`/etc/ssh2/sshd2_config:`
- `subsystem-sftp /usr/local/bin/sftp-server`
- The SFTP server is usually included as part of SSH packages from the public domain and is turned on by default configuration.
- SFTP is compatible with sftp server version OpenSSH_2.9.9p2 or higher.
- RSA-based user authentication is supported in the SSH and SFTP servers. The support, however, is not extended to the SSH client.
- Execution shell and SFTP are the only applications supported.
- The AES encryption algorithm is supported on the SSHv2 server and client, but not on the SSHv1 server and client. Any requests for an AES cipher sent by an SSHv2 client to an SSHv1 server are ignored, with the server using 3DES instead.
- The SFTP client does not support remote filenames containing wildcards (*, ?, []). The user must issue the sftp command multiple times or list all of the source files from the remote host to download them onto the router. For uploading, the router SFTP client can support multiple files specified using a wildcard provided that the issues mentioned in the first through third bullets in this section are resolved.
- The cipher preference for the SSH server follows the order AES128, AES192, AES256, and, finally, 3DES. The server rejects any requests by the client for an unsupported cipher, and the SSH session does not proceed.
- Use of a terminal type other than vt100 is unsupported, and the software generates a warning message in this case.
- Password messages of “none” are unsupported on the SSH client.
- Because the router infrastructure does not provide support for UNIX-like file permissions, files created on the local device lose the original permission information. For files created on the remote file system, the file permission adheres to the umask on the destination host and the modification and last access times are the time of the copy.

Information About Implementing Secure Shell

To implement SSH, you should understand the following concepts:

SSH Server

The SSH server feature enables an SSH client to make a secure, encrypted connection to a Cisco router. 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 IOS XR software authentication. The SSH server in Cisco IOS XR software works with publicly and commercially available SSH

clients.

SSH Client

The SSH client feature is an application running over the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco router to make a secure, encrypted connection to another Cisco router or to any other device running the SSH 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 secure communication over an insecure network. authentication. User authentication was performed in the Telnet session to the router. The user authentication mechanisms supported for SSH were RADIUS, TACACS+, and the use of locally stored usernames and passwords. The SSH client supports setting DSCP value in the outgoing packets.

```
ssh client dscp <value from 0 - 63>
```

- DSCP—DSCP value for SSH client sessions

```
RP/0/5/CPU0:router#configure
RP/0/5/CPU0:router(config)#ssh ?
  client    Provide SSH client service
  server    Provide SSH server service
  timeout   Set timeout value for SSH
RP/0/5/CPU0:router(config)#ssh client ?
```

- Knownhost—Enable the host pubkey check by the local database.
- Source-interface—Source interface for SSH client sessions

```
RP/0/5/CPU0:router(config)#ssh client source-interface ?
  ATM                ATM Network Interface(s)
  BVI                Bridge-Group Virtual Interface
  Bundle-Ether       Aggregated Ethernet interface(s)
  Bundle-POS         Aggregated POS interface(s)
  CEM                Circuit Emulation interface(s)
  GigabitEthernet    GigabitEthernet/IEEE 802.3 interface(s)
  IMA                ATM Network Interface(s)
  IMtestmain         IM Test Interface
  Loopback           Loopback interface(s)
  MgmtEth            Ethernet/IEEE 802.3 interface(s)
  Multilink          Multilink network interface(s)
  Null               Null interface
  PFItestmain        PFI Test Interface
  PFItestnothw       PFI Test Not-HW Interface
  POS                Packet over SONET/SDH network interface(s)
  PW-Ether           PWHE Ethernet Interface
  PW-IW              PWHE VC11 IP Interworking Interface
  Serial             Serial network interface(s)
  VASILeft           VASI Left interface(s)
  VASIRight          VASI Right interface(s)
  test-bundle-channel Aggregated Test Bundle interface(s)
  tunnel-ipsec       IPSec Tunnel interface(s)
  tunnel-mte         MPLS Traffic Engineering P2MP Tunnel interface(s)
  tunnel-te          MPLS Traffic Engineering Tunnel interface(s)
  tunnel-tp          MPLS Transport Protocol Tunnel interface
RP/0/5/CPU0:router(config)#ssh client source-interface
RP/0/5/CPU0:router(config)#
```

- VRF—Source interface VRF for SSH client sessions

```

RP/0/5/CPU0:router(config)#ssh client vrf ?
WORD   VRF name (max:32 chars)
RP/0/5/CPU0:router(config)#ssh client vrf shan ?
<cr>
RP/0/5/CPU0:router(config)#ssh client vrf shan

```

SSH also supports remote command execution as follows

```

RP/0/5/CPU0:router#ssh ?
A.B.C.D  IPv4 (A.B.C.D) address
WORD     Hostname of the remote node
X:X::X   IPv6 (A:B:C:D...:D) address
vrf      vrf table for the route lookup
RP/0/5/CPU0:router#ssh 1.1.1.1 ?
cipher          Accept cipher type
command         Specify remote command (non-interactive)
source-interface Specify source interface
username        Accept userid for authentication

```

```

<cr>
RP/0/5/CPU0:router#ssh 12.28.46.6 username admin command "show redundancy sum"
Password:

```

```

Wed Jan  9 07:05:27.997 PST
Active Node      Standby Node
-----
0/4/CPU0        0/5/CPU0 (Node Ready, NSR: Not Configured)

```

```
RP/0/5/CPU0:router#
```

SFTP Feature Overview

SSH includes support for standard file transfer protocol (SFTP) , a new standard file transfer protocol introduced in SSHv2. This feature provides a secure and authenticated method for copying router configuration or router image files. The SFTP client functionality is provided as part of the SSH component and is always enabled on the router. Therefore, a user with the appropriate level can copy files to and from the router. Like the copy command, the sftp command can be used only in EXEC mode. The SFTP client is VRF-aware, and you may configure the secure FTP client to use the VRF associated with a particular source interface during connections attempts. The SFTP client also supports interactive mode, where the user can log on to the server to perform specific tasks via the Unix server. The SFTP Server is a sub-system of the SSH server. In other words, when an SSH server receives an SFTP server request, the SFTP API creates the SFTP server as a child process to the SSH server. A new SFTP server instance is created with each new request.

The SFTP requests for a new SFTP server in the following steps:

- The user runs the sftp command with the required arguments
- The SFTP API internally creates a child session that interacts with the SSH server
- The SSH server creates the SFTP server child process
- The SFTP server and client interact with each other in an encrypted format
- The SFTP transfer is subject to LPTS policer "SSH-Known". Low policer values will affect SFTP transfer speeds

Note

In IOS-XR SW release 4.3.1 onwards, the default policer value for SSH-Known has been reset from 2500pps to 300pps. Slower transfers are expected due to this change. You can adjust the lpts policer value for this punt cause

to higher values that will allow faster transfers

When the SSH server establishes a new connection with the SSH client, the server daemon creates a new SSH server child process. The child server process builds a secure communications channel between the SSH client and server via key exchange and user authentication processes. If the SSH server receives a request for the subsystem to be an SFTP server, the SSH server daemon creates the SFTP server child process. For each incoming SFTP server subsystem request, a new SSH server child and an SFTP server instance is created. The SFTP server authenticates the user session and initiates a connection. It sets the environment for the client and the default directory for the user. Once the initialization occurs, the SFTP server waits for the SSH_FXP_INIT message from the client, which is essential to start the file communication session. This message may then be followed by any message based on the client's request. Here, the protocol adopts a 'request-response' model, where the client sends a request to the server; the server processes this request and sends a response. The SFTP server displays the following responses:

- Status Response
- Handle Response
- Data Response
- Name Response

Note The server must be running in order to accept incoming SFTP connections.

RSA-Based Host Authentication

Verifying the authenticity of a server is the first step to a secure SSH connection. This process is called host authentication and is conducted to ensure that a client connects to a valid server.

The host authentication is performed using the public key of a server. The server, during the key-exchange phase, provides its public key to the client. The client checks its database for known hosts of this server and the corresponding public key. If the client fails to find the server's IP address, it displays a warning message to the user, offering an option to either save the public key or discard it. If the server's IP address is found, but the public key does not match, the client closes the connection. If the public key is valid, the server is verified and a secure SSH connection is established.

The IOS XR SSH server and client had support for DSA-based host authentication. But for compatibility with other products, like IOS, RSA-based host authentication support is also added.

RSA-Based User Authentication

One of the methods for authenticating the user in SSH protocol is RSA public-key-based user authentication. The possession of a private key serves as the authentication of the user. This method works by sending a signature created with a private key of the user. Each user has an RSA keypair on the client machine. The private key of the RSA key pair remains on the client machine. The user generates an RSA public-private key pair on a Unix client using a standard key generation mechanism such as ssh-keygen. The max length of the keys supported is 2048 bits, and the minimum length is 512 bits. The following example displays a typical key generation activity:

- `bash-2.05b$ ssh-keygen -b 1024 -t rsa` Generating RSA private key, 1024-bit long modulus

The public key must be in base64 encoded (binary) format for it to be imported correctly into the box. You can use third-party tools available on the Internet to convert the key to the binary format. Once the public key is imported to the router, the SSH client can choose to use the public key authentication method by specifying the request using the "-o" option in the SSH client. For example:

- `client$ ssh -o PreferredAuthentications=publickey 1.2.3.4`

If a public key is not imported to a router using the RSA method, the SSH server initiates the password-based authentication. If a public key is imported, the server proposes the use of both the methods. The SSH client then chooses to use either method to establish the connection. The system allows only 10 outgoing SSH client connections. Currently, only SSH version 2 and SFTP servers support RSA-based authentication. For more information on how to import the public key to the router, see the Implementing Certification Authority Interoperability on the Cisco IOS XR Software chapter in this guide.

Note The preferred method of authentication would be as stated in the SSH RFC. The RSA-based authentication support is only for local authentication, and not for TACACS/RADIUS servers.

Authentication, Authorization, and Accounting (AAA) is a suite of network security services that provide the primary framework through which access control can be set up on your Cisco router or access server. For more information on AAA, see the Authentication, Authorization, and Accounting Commands on the Cisco IOS XR Software module in the Cisco IOS XR System Security Command Reference for the Cisco CRS Router publication and the Configuring AAA Services on the Cisco IOS XR Software Software module in the Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router publication.

SSHv2 Client Keyboard-Interactive Authentication

An authentication method in which the authentication information is entered using a keyboard is known as keyboard-interactive authentication. This method is an interactive authentication method in the SSH protocol. This type of authentication allows the SSH client to support different methods of authentication without having to be aware of their underlying mechanisms. Currently, the SSHv2 client supports the keyboard-interactive authentication. This type of authentication works only for interactive applications.

Note For SSHv1 configuration, Steps 1 to 4 are required. For SSHv2 configuration, Step 2 to Step 4 are optional. SSH server supports setting DSCP value in the outgoing packets.ssh server dscp <value from 0 – 63>

If not configured, the default DSCP value set in packets is 16 (for both client and server). This is the syntax for setting the DSCP value:

```
RP/0/5/CPU0:router(config)#ssh server dscp ?
    <0-63>  DSCP value range

RP/0/5/CPU0:router(config)#ssh server dscp 63 ?
    <cr>
RP/0/5/CPU0:router(config)#ssh server dscp 63
RP/0/5/CPU0:router(config)#

RP/0/5/CPU0:router(config)#ssh client dscp ?
    <0-63>  DSCP value range

RP/0/5/CPU0:router(config)#ssh client dscp 0 ?
    <cr>
RP/0/5/CPU0:router(config)#ssh client dscp 0
RP/0/5/CPU0:router(config)#
```

Perform this task to configure SSH.

SUMMARY STEPS

1. configure
2. hostname hostname
3. domain name domain-name
4. commit

5. crypto key generate rsa [usage keys | general-keys] [keypair-label]
6. crypto key generates DSA
7. configure
8. ssh timeout seconds
9. Do one of the following:
 1. ssh server [vrf vrf-name [ipv4 access-list IPv4 access-list name] [ipv6 access-list IPv6 access-list name]]
 2. ssh server v2
10. commit
11. show ssh
12. Show SSH session details

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	hostname <i>hostname</i> Example: RP/0/RP0/CPU0:router(config)# hostname router1	Configures a hostname for your router.

	Command or Action	Purpose
Step 3	domain name <i>domain-name</i> Example: RP/0/RP0/CPU0:router(config)# domain name cisco.com	Defines a default domain name that the software uses to complete unqualified host names.
Step 4	commit	
Step 5	crypto key generate rsa [usage keys general-keys] [<i>keypair-label</i>] Example: RP/0/RP0/CPU0:router# crypto key generate rsa general-keys	Generates an RSA key pair. • To delete the RSA key pair, use the crypto key zeroize rsa command. • This command is used for SSHv1 only.

Step 6	<p>crypto key generate dsa</p> <p>Example:</p> <p>RP/0/RP0/CPU0:router# crypto key generate dsa</p>	<p>Enables the SSH server for local and remote authentication on the router.</p> <ul style="list-style-type: none"> • The recommended minimum modulus size is 1024 bits. • Generates a DSA key pair. <p>To delete the DSA key pair, use the crypto key zeroize dsa command.</p> <ul style="list-style-type: none"> • This command is used only for SSHv2.
Step 7	<p>configure</p> <p>Example:</p> <p>RP/0/RP0/CPU0:router# configure</p>	<p>Enters global configuration mode.</p>
Step 8	<p>ssh timeout <i>seconds</i></p> <p>Example:</p> <p>RP/0/RP0/CPU0:router(config)# ssh timeout 60</p>	<p>(Optional) Configures the timeout value for user authentication to AAA.</p> <ul style="list-style-type: none"> • If the user fails to authenticate itself to AAA within the configured time, the connection is aborted. • If no value is configured, the default value of 30 seconds is used. The range is from 5 to 120.
Step 9	<p>Do one of the following:</p> <ul style="list-style-type: none"> • ssh server [vrf <i>vrf-name</i> [ipv4 access-list IPv4 access-list name] [ipv6 access-list IPv6 access-list name]] • ssh server v2 	<ul style="list-style-type: none"> • (Optional) Brings up an SSH server using a specified VRF of up to 32 characters. If no VRF is specified, the default VRF is used. To stop the SSH server from receiving any further connections for the specified VRF, use the no form of this command. If no VRF is specified, the default is assumed. Optionally ACLs for IPv4 and IPv6 can be used to restrict access to the server before the port is opened. To stop the SSH server from receiving any further connections for the

	Command or Action	Purpose
	<p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# ssh server vrf green ipv4 access-list list1 ipv6 access-list list 2</pre> <p>or</p> <pre>RP/0/RP0/CPU0:router(config)# ssh server v2</pre>	<p>specified VRF, use the no form of this command. If no VRF is specified, the default is assumed.</p> <p>Note The SSH server can be configured for multiple VRF usage.</p> <ul style="list-style-type: none"> • (Optional) Forces the SSH server to accept only SSHv2 clients if you configure the SSHv2 option by using the SSH server v2 command. If you choose the SSH server v2 command, only the SSH v2 client connections are accepted.
Step 10	commit	
Step 11	<p>show ssh</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show ssh</pre>	(Optional) Displays all of the incoming and outgoing SSH v1 and SSHv2 connections to the router.
Step 12	<p>show ssh session details</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show ssh session details</pre>	(Optional) Displays a detailed report of the SSHv2 connections to and from the router.

Configuring the SSH Client

Perform this task to configure an SSH client

SUMMARY STEPS

1. configure
2. ssh client known host device: /filename
3. commit
4. ssh [vrf vrf-name] {ipv4-address | ipv6-address | hostname} [**username user-id**] [**cipher aes { 128-cbc | 192-cbc | 256-cbc }**] source-interface type instance

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	

	Command or Action	Purpose
Step 2	<p>ssh client known host <i>device : /file name</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# ssh client known host slot1:/server_pubk ey</pre>	<p>(Optional) Enables the feature to authenticate and check the server public key (pubkey) at the client end.</p> <p>Note The complete path of the filename is required. The colon (:) and slash mark (/) are also required.</p>
Step 3	commit	
Step 4	<p>ssh [vrf vrf-name] { ipv4-address i pv6-address hostname } [usernam e user-id] [cipher aes { 128-cbc 192-cbc 256-cbc }] source-interface type i nstance]</p>	<p>Enables an outbound SSH connection.</p> <ul style="list-style-type: none"> To run an SSHv2 server, you must have a VRF. This may be the default or a specific VRF. VRF changes are applicable only to the SSH v2 server. The SSH client tries to make an SSHv2 connection to the remote peer. If the remote peer supports only the SSHv1 server, the peer internally spawns an SSHv1 connection to the remote server. The cipher des option can be used only with an SSHv1 client. The SSHv1 client supports only the 3DES encryption algorithm option, which is still available by default for those SSH clients only. If the <i>hostname</i> argument is used and the host has both IPv4 and IPv6 addresses, the IPv6 address is used.

- If you are using SSHv1 and your SSH connection is being rejected, you have not successfully generated an RSA key pair for your router. Make sure that you have specified a hostname and domain. Then use the crypto key generate RSA command to generate an RSA key pair and enable the SSH server.
- If you are using SSHv2 and your SSH connection is being rejected, you have not successfully generated a DSA key pair for your router. Make sure that you have specified a hostname and domain. Then use the crypto key generate dsa command to generate a DSA key pair and enable the SSH server.
- When configuring the RSA or DSA key pair, you might encounter the following error messages:
 - No hostname specified You must configure a hostname for the router using the hostname command.
 - domain specified You must configure a host domain for the router using the domain-name command.
- The number of allowable SSH connections is limited to the maximum number of virtual terminal lines configured for the router. Each SSH connection uses a vty resource.
- SSH uses either local security or the security protocol that is configured through AAA on your router for user authentication. When configuring AAA, you must ensure that the console is not running under AAA by applying

a keyword in the global configuration mode to disable AAA on the console.

Note

If you are using Putty version 0.63 or higher to connect to the SSH client, set the 'Chokes on PuTTYs SSH2 winadj request' option under SSH > Bugs in your Putty configuration to 'On.' This helps avoid a possible breakdown of the session whenever some long output is sent from IOS XR to the Putty client.

Configuration Examples for Implementing Secure Shell

This section provides the following configuration example:

Configuring Secure Shell: Example

This example shows how to configure SSHv2 by creating a hostname, defining a domain name, enabling the SSH server for local and remote authentication on the router by generating a DSA key pair, bringing up the SSH server, and saving the configuration commands to the running configuration file. After SSH has been configured, the SFTP feature is available on the router.

```
configure
hostname router1
domain name cisco.com
exit
crypto key generate dsa
configure
ssh server
end
```

Multi-channeling in SSH

The multi-channeling (also called multiplexing) feature on the Cisco IOS XR software server allows you to establish multiple channels over the same TCP connection. Thus, rather than opening a new TCP socket for each SSH connection, all the SSH connections are multiplexed into one TCP connection. For example, with multiplexing support on your XR software server, on a single SSH connection, you can simultaneously open a pseudo-terminal, remotely execute a command and transfer a file using any file transfer protocol. Multiplexing offers the following benefits:

- You are required to authenticate only once at the time of creating the session. After that, all the SSH clients associated with a particular session use the same TCP socket to communicate to the server.
- Saves time consumed otherwise wasted in creating a new connection each time.

Multiplexing is enabled by default in the Cisco IOS XR software server. If your client supports multiplexing, you must explicitly set up multiplexing on the client for it to be able to send multi-channel requests to the server. You can use OpenSSH, Putty, Perl, WinSCP, Putty, FileZilla, TTSSH, Cygwin or any other SSH-based tool to set up multiplexing on the client. Configure Client for Multiplexing provides an example of how you can configure the client for multiplexing using OpenSSH. For more information on the Multichannel feature, see the Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide, Release 5.1.1.

Restrictions for Multi-channeling Over SSH

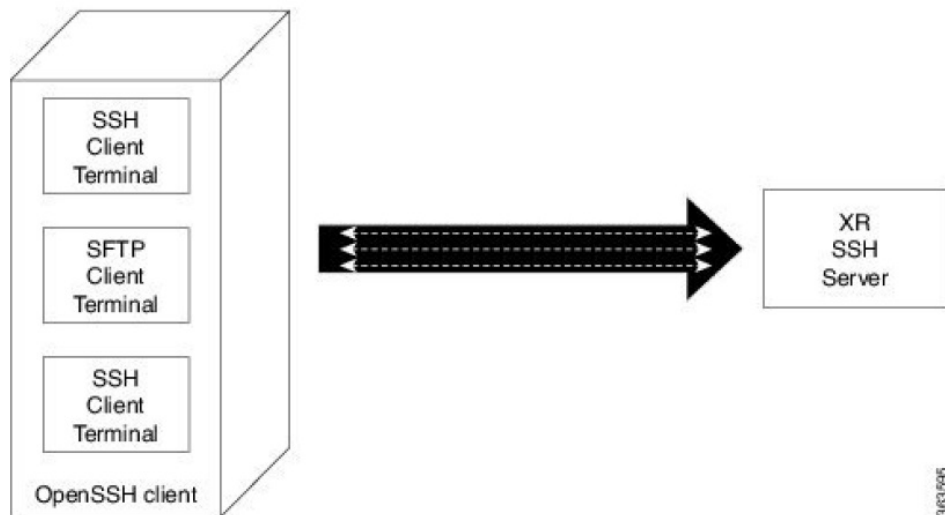
- Do not use client multiplexing for heavy transfer of data as the data transfer speed is limited by the TCP speed limit. Hence, for a heavy data transfer, it is advised that you run multiple SSH sessions, as the TCP speed limit

is per connection.

- Client multiplexing must not be used for more than 15 concurrent channels per session simultaneously.
- You must ensure that the first channel created at the time of establishing the session is always kept alive in order for other channels to remain open.

Client and Server Interaction Over Multichannel Connection

The figure below provides an illustration of a client-server interaction over an SSH multichannel connection.



As depicted in the illustration,

- The client multiplexes the collection of channels into a single connection. This allows different operations to be performed on different channels simultaneously. The dotted lines indicate the different channels that are open for a single session.
- After receiving a request from the client to open up a channel, the server processes the request. Each request to open up a channel represents the processing of a single service.

Note The Cisco IOX software supports server-side multiplexing only.

Configure Client for Multiplexing

The SSH client opens up one TCP socket for all the connections. In order to do so, the client multiplexes all the connections into one TCP connection. Authentication happens only once at the time of creating the session. After that, all the SSH clients associated with the particular session use the same TCP socket to communicate to the server. Use the following steps to configure client multiplexing using OpenSSH:

SUMMARY STEPS

1. Edit the `ssh_config` file.
2. Add entries `ControlMaster auto` and `ControlPath`
3. Create a temporary folder.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Edit the ssh_config file.	Open the ssh_config file with your favorite text editor to configure values for session multiplexing. The system-wide SSH configuration file is located under /etc/ssh/ssh_config. The user configuration file is located under ~/.ssh/config or \$HOME/.ssh/config.
Step 2	Add entries to ControlMaster auto	Add the entry ControlMaster auto and ControlPath to the ssh_config file, save it, and exit. <ul style="list-style-type: none"> ControlMaster determines whether SSH will listen for control connections and what to do about them. Setting the ControlMaster to 'auto' creates a master session automatically but if there is a master session already available, subsequent sessions are automatically multiplexed. ControlPath is the location for the control socket used by the multiplexed sessions. Specifying the ControlPath ensures that any time a connection to a particular server uses the same specified master connection.
	and ControlPath	
	Example: Host * ControlMaster auto ControlPath ~/.ssh/tmp/%r@%h:%p	
Step 3	Create a temporary folder.	Create a temporary directory inside the ~/.ssh folder for storing the control sockets.

Additional References

The following sections provide references related to implementing secure shell.

Related Documents

Related Topic	Document Title
AAA commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Authentication, Authorization, and Accounting Commands on the Cisco IOS XR Software module in Cisco IOS XR System Security Command Reference for the Cisco CRS Router.</i>
AAA configuration tasks	<i>Configuring AAA Services on the Cisco IOS XR Software module in the Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router.</i>
Host services and applications commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Host Services and Applications Commands on the Cisco IOS XR Software module in Cisco IOS XR IP Addresses and Services Command Reference for the Cisco CRS Router.</i>
IPSec commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>IPSec Network Security Commands on the Cisco IOS XR Software module in Cisco IOS XR System Security Command Reference for the Cisco CRS Router</i>
SSH commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Secure Shell Commands on the Cisco IOS XR Software module in Cisco IOS XR System Security Command Reference for the Cisco CRS Router</i>

Standards

Standards	Title
Draft-IETF-secs-user-auth-17.txt	<i>SSH Authentication Protocol, July 2003</i>
Draft-ietf-secsh-connect-17.txt	<i>SSH Connection Protocol, July 2003</i>
Draft-ietf-secsh-architecture-14.txt	<i>SSH Protocol Architecture, July 2003</i>
Draft-ietf-secsh-transport-16.txt	<i>SSH Transport Layer Protocol, July 2003</i>

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 6020	Netconf/ Yang

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Cisco IOS XR System Security Configuration Guide for the Cisco CRS Router, Release 5.2.x

Documents / Resources

	CISCO Implementing Secure Socket Layer [pdf] User Guide Implementing Secure Socket Layer, Secure Socket Layer, Socket Layer, Layer
-------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------

References

- [User Manual](#)