

NETCONF over SSHv2

You can use the Network Configuration Protocol (NETCONF) over Secure Shell Version 2 (SSHv2) feature to perform network configurations via the Cisco command-line interface (CLI) over an encrypted transport. The NETCONF Network Manager, which is the NETCONF client, must use Secure Shell Version 2 (SSHv2) as the network transport to the NETCONF server. Multiple NETCONF clients can connect to the NETCONF server.

- Prerequisites for NETCONF over SSHv2, on page 1
- Restrictions for NETCONF over SSH, on page 1
- Information About NETCONF over SSHv2, on page 2
- How to Configure NETCONF over SSHv2, on page 3
- Configuration Examples for NETCONF over SSHv2, on page 9
- Additional References for NETCONF over SSHv2, on page 11
- Feature Information for NETCONF over SSHv2, on page 12

Prerequisites for NETCONF over SSHv2

- NETCONF over SSHv2 requires that a vty line be available for each NETCONF session as specified in the **netconf max-session** command.

Restrictions for NETCONF over SSH

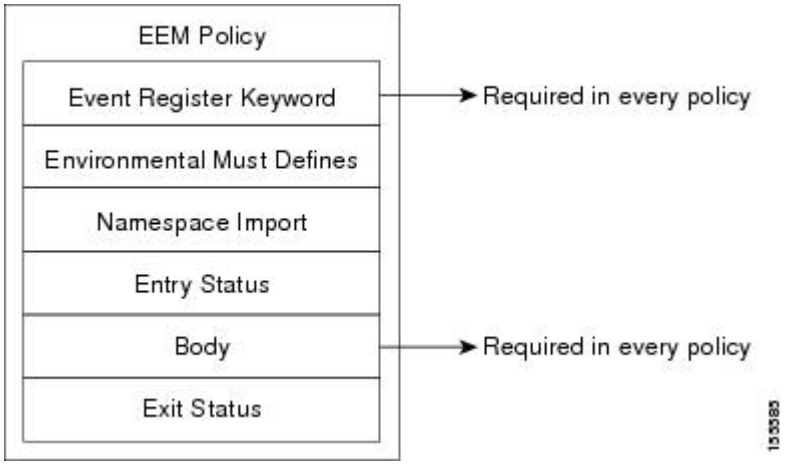
- Network Configuration Protocol (NETCONF) Secure Shell Version 2 (SSHv2) supports a maximum of 16 concurrent sessions.
- Only SSH version 2 is supported.

Information About NETCONF over SSHv2

NETCONF over SSHv2

To run the NETCONF over SSHv2 feature, the client (a Cisco device running Cisco software) establishes an SSH transport connection with the server (a NETCONF network manager). The following image shows a basic NETCONF over SSHv2 network configuration. The client and server exchange keys for security and password encryption. The user ID and password of the SSHv2 session running NETCONF are used for authorization and authentication purposes. The user privilege level is enforced and the client session may not have full access to the NETCONF operations if the privilege level is not high enough. If authentication, authorization, and accounting (AAA) is configured, the AAA service is used as if a user had established an SSH session directly to the device. Using the existing security configuration makes the transition to NETCONF almost seamless. Once the client has been successfully authenticated, the client invokes the SSH connection protocol and the SSH session is established. After the SSH session is established, the user or application invokes NETCONF as an SSH subsystem called “netconf.”

Figure 1: NETCONF over SSHv2



Secure Shell Version 2

SSHv2 runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. SSHv2 provides a means to securely access and securely execute commands on another computer over a network.

NETCONF does not support SSH version 1. The configuration for the SSH Version 2 server is similar to the configuration for SSH version 1. Use the **ip ssh version** command to specify 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 device to fall back to the undefined protocol (version 1), you should use the **ip ssh version** command and specify version 2.

Use the **ip ssh rsa keypair-name** command to enable an SSH connection using Rivest, Shamir, and Adelman (RSA) keys that you have configured. 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 do not need to configure a hostname and a domain name.

How to Configure NETCONF over SSHv2

Enabling SSH Version 2 Using a Hostname and Domain Name

Perform this task to configure your device for SSH version 2 using a hostname and domain name. You may also configure SSH version 2 by using the RSA key pair configuration (see Enabling SSH Version 2 Using RSA Key Pairs, on page 4).

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 2**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *hostname* **Example:** `Device(config)# hostname host1`	Configures a hostname for your device.
Step 4	**ip domain-name** *name* **Example:**	Configures a domain name for your device.

	Command or Action	Purpose
	`Device(config)# ip domain-name domain1.com`	
Step 5	**crypto key generate rsa** **Example:** `Device(config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication.
Step 6	**ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*] **Example:** `Device(config)# ip ssh timeout 120`	(Optional) Configures SSH control variables on your device.
Step 7	**ip ssh version 2** **Example:** `Device(config)# ip ssh version 2`	Specifies the version of SSH to be run on your device.

Enabling SSH Version 2 Using RSA Key Pairs

Perform this task to enable SSH version 2 without configuring a hostname or domain name. 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 hostname and domain name configuration. (See "Enabling SSH Version 2 Using a Hostname and Domain Name, on page 3.)

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 2**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip ssh rsa keypair-name** *keypair-name* **Example:** `Device(config)# ip ssh rsa keypair-name sshkeys`	Specifies which RSA keypair to use for SSH usage. **Note** A Cisco device can have many RSA key pairs.
Step 4	**crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size* **Example:** `Device(config)# crypto key generate rsa usage-keys label sshkeys modulus 768`	Enables the SSH server for local and remote authentication on the device. For SSH version 2, the modulus size must be at least 768 bits. **Note** To delete the RSA key pair, use the **crypto key zeroize rsa** command. After you have deleted the RSA command, you automatically disable the SSH server.
Step 5	**ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*] **Example:** `Device(config)# ip ssh timeout 120`	Configures SSH control variables on your device.
Step 6	**ip ssh version 2** **Example:** `Device(config)# ip ssh version 2`	Specifies the version of SSH to be run on a device.

Starting an Encrypted Session with a Remote Device

Perform this task to start an encrypted session with a remote networking device. (You do not have to enable your device. SSH can be run in disabled mode.)

From any UNIX or UNIX-like device, the following command is typically used to form an SSH session:

```
ssh -2 -s user@router.example.com netconf
```

SUMMARY STEPS

1. Do one of the following:
   - **ssh** [**-v** {**1** | **2**}] [**-c** {**3des**| **aes128-cbc** | **aes192-cbc**| **aes256-cbc**}] [**-m**{**hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96**}] [**l** *userid*] [**-o numberofpasswordprompts** *n*] [**-p** *port-num*] {*ip-addr* | *hostname*} [*command*]

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	Do one of the following:
	• **ssh** [**-v** {**1** | **2**}] [**-c** {**3des**| **aes128-cbc** | **aes192-cbc**| **aes256-cbc**}] [**-m**{**hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96**}] [**l** *userid*] [**-o numberofpasswordprompts** *n*] [**-p** *port-num*] {*ip-addr* | *hostname*} [*command*]
	Example:
	`Device# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 -l user2 10.76.82.24`
	Example:
	`Device# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 user2@10.76.82.24`
		Starts an encrypted session with a remote networking device.
		The first 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 second configuration example provides an end result that is identical to that of the first example.

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.

What to Do Next

For more information about the **ssh** command, see the Cisco IOS Security Command Reference.

Verifying the Status of the Secure Shell Connection

Perform this task to display the status of the SSH connection on your device.

Note You can use the following **show** commands in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show ssh**
3. **show ip ssh**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ssh** **Example:** `Device# show ssh`	Displays the status of SSH server connections.
Step 3	**show ip ssh** **Example:** `Device# show ip ssh`	Displays the version and configuration data for SSH.

Examples

The following output from the **show ssh** command displays status about SSH version 2 connections.

```
Device# 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.
```

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

```
Device# show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
```

Enabling NETCONF over SSHv2

Perform this task to enable NETCONF over SSHv2.

Before you begin

SSHv2 must be enabled.

Note There must be at least as many vty lines configured as there are concurrent NETCONF sessions.

Note
- A minimum of four concurrent NETCONF sessions must be configured.
- A maximum of 16 concurrent NETCONF sessions can be configured.
- NETCONF does not support SSHv1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **netconf ssh** [**acl** *access-list-number*]
4. **netconf lock-time** *seconds*
5. **netconf max-sessions** *session*
6. **netconf max-message** size

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**netconf ssh** [**acl** *access-list-number*] **Example:** `Device(config)# netconf ssh acl 1`	Enables NETCONF over SSHv2. • Optionally, you can configure an access control list for this NETCONF session.
Step 4	**netconf lock-time** *seconds* **Example:** `Device(config)# netconf lock-time 60`	(Optional) Specifies the maximum time, in seconds, a NETCONF configuration lock is in place without an intermediate operation. • The valid range is 1 to 300. The default value is 10 seconds.

	Command or Action	Purpose
Step 5	**netconf max-sessions** *session* **Example:** `Device(config)# netconf max-sessions 5`	(Optional) Specifies the maximum number of concurrent NETCONF sessions allowed. • The valid range is 4 to 16. The default value is 4.
Step 6	**netconf max-message** size **Example:** `Device(config)# netconf max-message 37283`	(Optional) Specifies the maximum size, in kilobytes (KB), for the messages received in a NETCONF session. • The valid range is 1 to 2147483. The default value is infinite. • To set the maximum size to infinite, use the **no netconf max-message** command.

Configuration Examples for NETCONF over SSHv2

Example: Enabling SSHv2 Using a Hostname and Domain Name

```
configure terminal
hostname host1
ip domain-name example.com
crypto key generate rsa
ip ssh timeout 120
ip ssh version 2
```

Enabling Secure Shell Version 2 Using RSA Keys Example

The following example shows how to configure SSHv2 using RSA keys:

```
Device# configure terminal

Device(config)# ip ssh rsa keypair-name sshkeys

Device(config)# crypto key generate rsa usage-keys label sshkeys modulus 768
Device(config)# ip ssh timeout 120
Device(config)# ip ssh version 2
```

Starting an Encrypted Session with a Remote Device Example

The following example shows how to start an encrypted SSH session with a remote networking device, from any UNIX or UNIX-like device:

```
Device(config)# ssh -2 -s user@router.example.com netconf
```

Configuring NETCONF over SSHv2 Example

The following example shows how to configure NETCONF over SSHv2:

```
Device# configure terminal
Device(config)# netconf ssh acl 1
Device(config)# netconf lock-time 60
Device(config)# netconf max-sessions 5
Device(config)# netconf max-message 2345
Device# ssh-2 -s username@10.1.1.1 netconf
```

The following example shows how to get the configuration for loopback interface 113.

SUMMARY STEPS

1. First, send the “hello”:
2. Next, send the get-config request:

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	First, send the “hello”:
	Example:
	`<?xml version="1.0" encoding=\"UTF-8\"?>`
	`<hello><capabilities>`
	`<capability>u?rn:ietf:params:netconf:base:1.0</capability>`
	`<capability>urn:ietf:params:netconf:capability:writeable-running:1.0</capability>`
	`<capability>urn:ietf:params:netconf:ca?pability:roll?back-on-error:1.0</capability>`
	`<capability>urn:ietf:params:netconf:capability:startup:1.0</capability>`
	`<capability>urn:ietf:params:netconf:ca?pability:url:?1.0</capability>`
	`<capability>urn:cisco:params:netconf:capability:pi-data-model:1.0</capability>`
	`<capability>urn:cisco:params:netconf:capabili?ty:notificati?on:1.0</capability>`
	`</capabilities>`
	`</hello>]]>]]>`	

Step 2	Next, send the get-config request: **Example:**	

	Command or Action	Purpose
	<?xml version="1.0"?> <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"xmlns:cpi="http://www.cisco.com/cpi_10/schema" message-id="101"> <get-config> <source> <running/> </source> <filter> <config-format-text-cmd> <text-filter-spec> interface Loopback113 </text-filter-spec> </config-format-text-cmd> </filter> </get-config> </rpc>]]>]]>	

The following output is shown on the device:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101"xmlns=\"urn:ietf:params:netconf:base:1.0\">
   <data>
      <cli-config-data>
interface Loopback113
description test456
no ip address
load-interval 30
end
       </cli-config-data>
    </data>
</rpc-rep?ly>]]>]]>
```

Additional References for NETCONF over SSHv2

Related Documents

Related Topic	Document Title
Cicso IOS Commands	Cisco IOS Master Command List, All Releases
NETCONF commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Cisco Networking Services Command Reference*
IP access lists commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples Security commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*

Related Topic	Document Title
IP access lists	IP Access List Overview and Creating an IP Access List and Applying It to an Interface modules in the Cisco IOS Security Configuration Guide: Securing the Data Plane.
Secure Shell and Secure Shell Version 2	"Configuring Secure Shell" module in the Cisco IOS Security Configuration Guide: Securing User Services.

Standards and RFCs

RFC	Title
RFC 2246	*The TLS Protocol Version 1.0*
RFC 4251	*The Secure Shell (SSH) Protocol Architecture*
RFC 4252	*The Secure Shell (SSH) Authentication Protocol*
RFC 4741	NETCONF Configuration Protocol
RFC 4742	Using the NETCONF Configuration Protocol over Secure SHell (SSH)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for NETCONF over SSHv2

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 1: Feature Information for NETCONF over SSHv2

Feature Name	Releases	Feature Information
NETCONF over SSHv2	Cisco IOS XE Release 2.1 12.2(33)SB 12.2(33)SRA 12.2(33)SXI 12.4(9)T	The NETCONF over SSHv2 feature enables you to perform network configurations via the Cisco command-line interface (CLI) over an encrypted transport. The following commands were introduced or modified by this feature: **netconf lock-time**, **netconf max-message**, **netconf max-sessions** **netconf ssh**.