NETCONF over SSHv2

First Published: June 19, 2006
Last Updated: June 19, 2006

The NETCONF over SSHv2 feature enables you to perform network configurations via Cisco command-line interface (CLI) over an encrypted transport.

The network configuration protocol (NETCONF) defines a simple mechanism through which a network device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated. NETCONF uses Extensible Markup Language (XML)-based data encoding for the configuration data and protocol messages.

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.

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 NETCONF over SSHv2" section on page 30.

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

CISCO SYSTEMS

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

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 Access over SSHv2

- Only SSH version 2 is supported.
- NETCONF SSHv2 supports a maximum of 16 concurrent sessions.

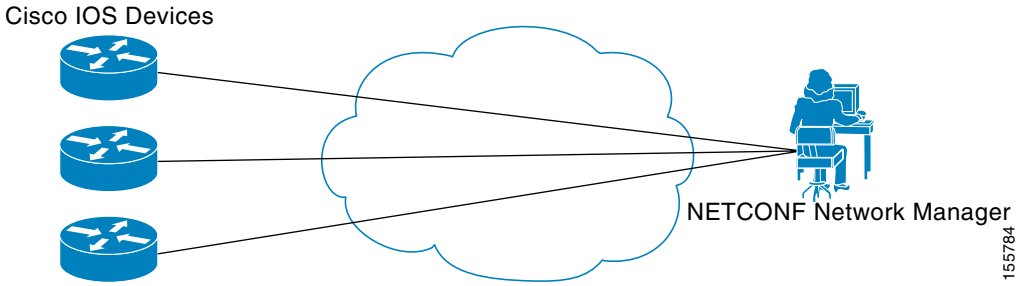
Information About NETCONF over SSHv2

To configure the NETCONF over SSHv2 feature, you should understand the following concepts:

NETCONF over SSHv2

To run the NETCONF over SSHv2 feature, the client (a Cisco device running Cisco IOS software) establishes an SSH transport connection with the server (a NETCONF Network Manager.) Figure 1 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



NETCONF Notifications

NETCONF sends notifications of any configuration change over NETCONF. A notification is an event indicating that a configuration change has occurred. The change can be a new configuration, deleted configuration, or changed configuration. The notifications are sent at the end of a successful configuration operation as one message that shows the set of changes rather than showing individual messages for each line in the configuration that is changed.

Secure Shell Version 2

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 router 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.

Access Lists

You can optionally configure access lists for use with NETCONF over SSHv2 sessions. An access list is a sequential collection of permit and deny conditions that apply to IP addresses. The Cisco IOS software tests addresses against the conditions in an access list one by one. The first match determines whether the software accepts or rejects the address. Because the software stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the software rejects the address.

The two main tasks involved in using access lists are as follows:

1. Creating an access list by specifying an access list number or name and access conditions.
2. Applying the access list to interfaces or terminal lines.

For more information about configuring access lists, see the "IP Access Lists" section of the *Cisco IOS IP Application Services Configuration Guide*, Release 12.4.

How to Configure NETCONF over SSHv2

This section contains the following tasks:

Configuring a Router for SSH Version 2 Using a HostName and Domain Name

To configure your router for SSH version 2 using a hostname and domain name, perform the following steps. You may also configure SSH version 2 by using the RSA key pair configuration (See "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 2**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *hostname* **Example:** `Router(config)# hostname host1`	Configures a hostname for your router.
Step 4	**ip domain-name** *name* **Example:** `Router(config)# ip domain-name domain1.com`	Configures a domain name for your router.
Step 5	**crypto key generate rsa** **Example:** `Router(config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication.
Step 6	**ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*] **Example:** `Router(config)# ip ssh timeout 120`	(Optional) Configures SSH control variables on your router.
Step 7	**ip ssh version 2** **Example:** `Router(config)# ip ssh version 2`	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 hostname 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 hostname and domain name configuration. (See "Configuring a Router for SSH Version 2 Using a HostName 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 2**

DETAILED STEPS

Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip ssh rsa keypair-name** *keypair-name* **Example:** `Router(config)# ip ssh rsa keypair-name sshkeys`	Specifies which RSA keypair to use for SSH usage. **Note** A Cisco IOS router can have many RSA key pairs.
Step 4	**crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size* **Example:** `Router(config)# crypto key generate rsa usage-keys label sshkeys modulus 768`	Enables the SSH server for local and remote authentication on the router. 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:** `Router(config)# ip ssh timeout 120`	Configures SSH control variables on your router.
Step 6	**ip ssh version 2** **Example:** `Router(config)# ip ssh version 2`	Specifies the version of SSH to be run on a router.

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.)

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. **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

Step 1	**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:
	`Router# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 -l user2 10.76.82.24`
	
	or
	
	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:
	
	`Router# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 user2@10.76.82.24`
		Starts an encrypted session with a remote networking device.

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*, Release 12.2SR.

Verifying the Status of the Secure Shell Connection

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

SUMMARY STEPS

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

DETAILED STEPS

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

Examples

The following output examples from the **show ssh** and **show ip ssh** commands display status about SSH version 2 connections.

```
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.
```

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.

```
Router# 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.

Prerequisites

- SSHv2 must be enabled.

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

Restrictions

- 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*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**netconf ssh** [**acl** *access-list-number*] Example: `Router(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:** `Router(config)# netconf lock-time 60`	(Optional) Specifies the maximum time a NETCONF configuration lock is in place without an intermediate operation. The valid range is 1 to 300 seconds. The default value is 10 seconds.
Step 5	**netconf max-sessions** *session* **Example:** `Router(config)# netconf max-sessions 5`	(Optional) Specifies the maximum number of concurrent NETCONF sessions allowed.

Configuring the NETCONF Network Manager Application

Use the following CLI string to configure the NETCONF Network Manager application to invoke NETCONF as an SSH subsystem:

```
Unix Side: ssh-2 -s cisco@10.1.1.1 netconf
```

Use the following XML string to enable the NETCONF Network Manager application to send and receive NETCONF notifications:

```
<?xml version="1.0" encoding="UTF-8" ?><rpc message-id="9.0"><notification-on/></rpc>
```

Use the following XML string to stop the NETCONF Network Manager application from sending or receiving NETCONF notifications:

```
<?xml version="1.0" encoding="UTF-8" ?><rpc message-id="9.13"><notification-off/></rpc>
```

Use the following XML to deliver the NETCONF payload to the Network Manager application:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.cisco.com/cpi_10/schema"
elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns="http://www.cisco.com/cpi_10/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <!--The following elements define the cisco extensions for the content of the filter
element in a <get-config> request. They allow the client to specify the format of the
response and to select subsets of the entire configuration to be included.-->
   <xs:element name="config-format-text-block">
      <xs:annotation>
         <xs:documentation>If this element appears in the filter, then the cllient is
requesting that the response data be sent in config command block
format.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="text-filter-spec" minOccurs="0"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="config-format-text-cmd">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="text-filter-spec"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="config-format-xml">
      <xs:annotation>
         <xs:documentation>When this element appears in the filter of a get-config
request, the results are to be returned in E-DI XML format. The content of this element is
treated as a filter.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:complexContent>
            <xs:extension base="xs:anyType"/>
         </xs:complexContent>
      </xs:complexType>
   </xs:element>
   <!--These elements are used in the filter of a <get> to specify operational data to
return.-->
   <xs:element name="oper-data-format-text-block">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="show" type="xs:string" maxOccurs="unbounded"/>
```

```
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="oper-data-format-xml">
      <xs:complexType>
         <xs:sequence>
            <xs:any/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <!--When confing-format-text format is specified, the following describes the content
of the data element in the response-->
   <xs:element name="cli-config-data">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="cmd" type="xs:string" maxOccurs="unbounded">
               <xs:annotation>
                  <xs:documentation>Content is a command. May be multiple
lines.</xs:documentation>
               </xs:annotation>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="cli-config-data-block" type="xs:string">
      <xs:annotation>
         <xs:documentation>The content of this element is the device configuration as it
would be sent to a terminal session. It contains embedded newline characters that must be
preserved as they represent the boundaries between the individual command
lines</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:element name="text-filter-spec">
      <xs:annotation>
         <xs:documentation>If this element is included in the config-format-text element,
then the content is treated as if the string was appended to the "show running-config"
command line.</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:element name="cli-oper-data-block">
      <xs:complexType>
         <xs:annotation>
            <xs:documentation> This element is included in the response to get operation.
Content of this element is the operational data in text format.</xs:documentation>
         </xs:annotation>
         <xs:sequence>
            <xs:element name="item" maxOccurs="unbounded">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="show"/>
                     <xs:element name="response"/>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="xml-oper-data">
      <xs:complexType>
         <xs:annotation>
            <xs:documentation> This element is included in the response to get operation.
Content of this element is the operational data in xml format.</xs:documentation>
         </xs:annotation>
         <xs:sequence>
```

```
            <xs:any/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="xml-config-data">
      <xs:complexType>
         <xs:annotation>
            <xs:documentation> This element is included in the response to get-config and
get operations. Content of this element is the configuration data in xml
format.</xs:documentation>
         </xs:annotation>
         <xs:sequence>
            <xs:any/>
         </xs:sequence>
      </xs:complexType>
```

Formatting NETCONF Notifications

The NETCONF Network Manager application uses .xsd schema files to describe the format of the XML NETCONF notification messages being sent between a NETCONF Network Manager application and a router running NETCONF over SSHv2. These files can be displayed in a browser or a schema reading tool. You can use these schema to validate that the XML is correct. These schema describe the format, not the content, of the data being exchanged.

NETCONF uses the <edit-config> function to load all of a specified configuration to a specified target configuration. When this new configuration is entered, the target configuration is not replaced. The target configuration is changed according to the data and requested operations of the requesting source.

The following are schemas for the NETCONF <edit-config> function in CLI, CLI block, and XML format:

NETCONF <edit-config> Request: CLI Format

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:cpi="http://www.cisco.com/cpi_10/schema">
   <edit-config>
      <target>
         <running/>
      </target>
      <config>
         <cpi:config-format-xml xmlns="http://www.cisco.com/edi_20/Cat3550/12.1">
            <Hostname>test</Hostname>
            <Interface>
               <InterfaceName>fastEthernet0/1</InterfaceName>
               <IP>
                  <Address>
                     <IPAddress>192.168.1.1</IPAddress>
                     <Mask>255.255.255.0</Mask>
                  </Address>
               </IP>
            </Interface>
         </cpi:config-format-xml>
      </config>
   </edit-config>
</rpc>
```

NETCONF <edit-config> Response: CLI Format

```
<rpc-reply>
   <ok/>
```

```
</rpc-reply>
```

NETCONF <edit-config> Request: CLI-Block Format

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:cpi="http://www.cisco.com/cpi_10/schema">
   <edit-config>
      <target>
         <running/>
      </target>
      <config>
         <cpi:config-format-ios-text-block>
            hostname test
            interface fastEthernet0/1
            ip address 192.168.1.1 255.255.255.0
         </cpi:config-format-ios-text-block>
      </config>
   </edit-config>
</rpc>
```

NETCONF <edit-config> Response: CLI-Block Format

```
<rpc-reply>
   <ok/>
</rpc-reply>
```

NETCONF <edit-config> Request: XML Format

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:cpi="http://www.cisco.com/cpi_10/schema">
   <edit-config>
      <target>
         <running/>
      </target>
      <config>
         <cpi:config-format-xml xmlns="http://www.cisco.com/edi_20/Cat3550/12.1">
            <Hostname>test</Hostname>
            <Interface>
               <InterfaceName>fastEthernet0/1</InterfaceName>
               <IP>
                  <Address>
                     <IPAddress>192.168.1.1</IPAddress>
                     <Mask>255.255.255.0</Mask>
                  </Address>
               </IP>
            </Interface>
         </cpi:config-format-xml>
      </config>
   </edit-config>
</rpc>
```

NETCONF <edit-config> Response: XML Format

```
<rpc-reply>
   <ok/>
</rpc-reply>
```

NETCONF uses the <get-config> function to retrieve all or part of a configuration. The <source> element is the name of the configuration database being queried. The <filter> element identifies the portions of the device configuration to retrieve. If the <filter> element is empty or unspecified, the entire configuration is returned.

The following are schemas for the NETCONF <get-config> function in CLI and CLI-block format:

NETCONF <get-config> Request: CLI Format

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:cpi="http://www.cisco.com/cpi_10/schema">
   <get-config>
      <source>
         <running/>
      </source>
      <filter>
         <cpi:config-format-ios-text-cmd>
            <cpi:ios-text-filter-spec> | interface </cpi:ios-text-filter-spec>
         </cpi:config-format-ios-text-cmd>
</filter>
   </get-config>
</rpc>
```

NETCONF <get-config> Response: CLI Format

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:cpi="http://www.cisco.com/cpi_10/schema">
   <data>
      <cpi:cli-data>
         <cpi:cmd>interface fastEthernet0/1</cpi:cmd>
         <cpi:cmd>interface fastEthernet0/2</cpi:cmd>
      </cpi:cli-data>
   </data>
</rpc-reply>
```

NETCONF <get-config> Request: CLI-Block Format

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:cpi="http://www.cisco.com/cpi_10/schema">
   <get-config>
      <source>
         <running/>
      </source>
      <filter>
         <cpi:config-format-ios-text-block>
            <cpi:ios-text-filter-spec> | interface </cpi:ios-text-filter-spec>
         </cpi:config-format-ios-text-block>
      </filter>
   </get-config>
</rpc>
```

NETCONF <get-config> Response: CLI-Block Format

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:cpi="http://www.cisco.com/cpi_10/schema">
   <data>
      <cpi:cli-data-block>
         interface fastEthernet0/1
         interface fastEthernet0/2
      </cpi:cli-data-block>
   </data>
</rpc-reply>
```

NETCONF uses the <get> function to retrieve configuration and device-state information. The NETCONF <get> format is the equivalent of a Cisco IOS **show** command. The <filter> parameter specifies the portion of the system configuration and device-state data to retrieve. If the <filter> parameter is empty, nothing is returned.

The following are schemas for the <get> function in CLI and CLI-block format:

NETCONF <get> Request: CLI Format

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:cpi="http://www.cisco.com/cpi_10/schema">
    <get>
       <filter>
          <cpi:config-format-cli-ios-text-cmd>
             <ios-filter-text-spec> | include interface </ios-filter-text-spec>
          </cpi:config-format-cli-ios-text-cmd>
          <cpi:cli-operational-data>
             <show>interfaces</show>
             <show>arp</show>
          </cpi:cli-operational-data>
       </filter>
    </get>
 </rpc>
```

NETCONF <get> Response: CLI Format

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:cpi="http://www.cisco.com/cpi_10/schema">
    <data>
       <cpi:cli-data>
          <cpi:cmd>interface fastethernet0/1</cpi:cmd>
          <cpi:cmd>interface loopback</cpi:cmd>
          <cpi:cmd>interface gigabit</cpi:cmd>
       </cpi:cli-data>
       <cpi:cli-operational-data>
          <item>
             <show>interfaces</show>
             <response>
                <!-- output of "show interfaces" ------>
             </response>
          <show>arp</show>
          <item>
             <show>arp</show>
             <response>
                <!-- output of "show arp" ------>
             </response>
          </item>
       </cpi:cli-operational-data>
    </data>
</rpc-reply>
```

NETCONF <get> Request: CLI-Block Format

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:cpi="http://www.cisco.com/cpi_10/schema">
    <get>
       <filter>
          <cpi:config-format-cli-ios-text-block>
             <ios-filter-text-spec> | include interface </ios-filter-text-spec>
          </cpi:config-format-cli-ios-text-block>
          <cpi:cli-operational-data>
             <show>interfaces</show>
             <show>arp</show>
          </cpi:cli-operational-data>
       </filter>
    </get>
 </rpc>
```

NETCONF <get> Response: CLI-Block Format

```
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:cpi="http://www.cisco.com/cpi_10/schema">
    <data>
       <cpi:cli-data-block>
          interface fastethernet0/1
          interface loopback
          interface gigabit
       </cpi:cli-data-block>
       <cpi:cli-operational-data>
          <item>
             <show>interfaces</show>
             <response>
                <!-- output of "show interfaces" ------>
             </response>
          <show>arp</show>
          <item>
             <show>arp</show>
             <response>
                <!-- output of "show arp" ------>
             </response>
          </item>
       </cpi:cli-operational-data>
    </data>
</rpc-reply>
```

Monitoring and Maintaining NETCONF Sessions

Perform this task to monitor and maintain NETCONF sessions.

Prerequisites

SSHv2 must be enabled.

Restrictions

- 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. **show netconf** {**counters** | **session**}
3. **debug netconf** {**all** | **error**}
4. **clear netconf** {**counters** | **sessions**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show netconf** {**counters** | **session**} **Example:** `Router# show netconf counters`	Clears NETCONF statistics counters, NETCONF sessions, and to free associated resources and locks.
Step 3	**debug netconf** {**all** | **error**} **Example:** `Router# debug netconf error`	Enables debugging of NETCONF sessions.
Step 4	**clear netconf** {**counters** | **sessions**} **Example:** `Router# clear netconf sessions`	Clears NETCONF statistics counters, NETCONF sessions, and to free associated resources and locks.

Configuration Examples for NETCONF over SSHv2

This section provides the following configuration examples:

Configuring SSHv2 Using a Host Name and Domain Name

The following example shows how to configure SSHv2 using a hostname and a domain name:

```
configure terminal
hostname host1
ip domain-name domain1.com
crypto key generate rsa
ip ssh timeout 120
ip ssh version 2
```

Configuring Secure Shell Version 2 Using RSA Keys: Example

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

```
configure terminal
ip ssh rsa keypair-name sshkeys
crypto key generate rsa usage-keys label sshkeys modulus 768
```

```
ip ssh timeout 120
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:

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

Configuring NETCONF over SSHv2: Example

The following example shows how to configure NETCONF over SSHv2:

```
configure terminal
netconf ssh acl 1
netconf lock-time 60
netconf max-sessions 5
```

Additional References

The following sections provide references related to NETCONF over SSHv2.

Related Documents

Related Topic	Document Title
Secure Shell and Secure Shell Version 2	"Configuring Secure Shell" and "Configuring Secure Shell Version 2 Support" sections of the *Cisco IOS Security Configuration Guide*, Release 12.4.
Security Commands	*Cisco IOS Security Command Reference*, Release 12.4.
IP Access Lists	"Configuring IP Access Lists" section of the *Cisco IOS IP Application Services Configuration Guide*, Release 12.4.
IP Access Lists Commands	*Cisco IOS IP Application Services Command Reference*, Release 12.2 SR.
NETCONF over BEEP	*NETCONF over BEEP*

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 4251	*The Secure Shell (SSH) Protocol Architecture*
RFC 4252	*The Secure Shell (SSH) Authentication Protocol*

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation 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

Command Reference

This section documents new commands only.

- **clear netconf**
- **debug netconf**
- **netconf lock-time**
- **netconf max-sessions**
- **netconf ssh**
- **show netconf**

clear netconf

To clear network configuration protocol (NETCONF) statistics counters or NETCONF sessions and to free associated resources and locks, use the **clear netconf** command in privileged EXEC mode.

clear netconf {**counters** | **sessions**}

Syntax Description

counters	Clears the NETCONF statistics counters to zero.
sessions	Clears currently connected NETCONF sessions.

Command Default

NETCONF statistics counters are incremented and configured NETCONF sessions remain active.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRA	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines

Use this command to clear NETCONF statistics counters to zero, to clear all or specified NETCONF sessions and to disconnect and free associated resources and locks.

Examples

The following example shows how to clear all NETCONF counters:

```
clear netconf counters
```

Related Commands

Command	Description
debug netconf	Enables debugging of NETCONF sessions.
netconf lock-time	Specifies the maximum time a NETCONF configuration lock is in place without an intermediate operation.
netconf max-sessions	Specifies the maximum number of concurrent NETCONF sessions allowed.
netconf ssh	Enables NETCONF over SSHv2.
show netconf	Displays NETCONF statistics counters and session information.

debug netconf

To enable debugging of network configuration protocol (NETCONF) sessions, use the **debug netconf** command in privileged EXEC mode. To turn off NETCONF debugging, use the **no** form of this command.

debug netconf {**all** | **error**}

no debug netconf {**all** | **error**}

Syntax Description

all	Enables debugging of NETCONF sessions, including NETCONF errors.
error	Enables debugging of NETCONF errors.

Command Default

NETCONF debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRA	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines

The **debug netconf** command issues debug information only when an operational error has happened. In most situations, the NETCONF notifications sent between the NETCONF Network Manager and the client are sufficient to diagnose most NETCONF problems.

To view Extensible Markup Language (XML) parsing errors when using NETCONF over SSHv2, you must also configure the **debug cns xml all** command.

Examples

The following example shows how to enable debugging of all NETCONF sessions:

```
Router# debug netconf

00:14:03: NETCONF-ERROR: could not find user1
00:14:03: NETCONF-ERROR: could not find tftp://samplelocation/samplefile
00:14:03: NETCONF: locking 1 by session 646B7038
00:14:03: NETCONF: locking 2 by session 646B7038
00:14:03: NETCONF: locking 1 by session 646B7038
00:14:03: NETCONF-ERROR: invalid session unlock attempt
00:14:03: NETCONF: locking 1 by session 646B7038
00:14:03: NETCONF-ERROR: lock already active
00:14:13: NETCONF-ERROR: lock time 1 expired closing session 646B7038
```

Table 1 describes the significant fields shown in the display.

Table 1 debug netconf Field Descriptions

Field	Description
NETCONF-ERROR: could not find user1	NETCONF could not find the specified username.
NETCONF-ERROR: could not find tftp://samplelocation/samplefile	NETCONF could not find the specified file path.
NETCONF: locking 1 by session 646B7038	This user is locking NETCONF.
NETCONF-ERROR: invalid session unlock attempt	Another user is trying to unlock NETCONF without first acquiring the lock.
NETCONF-ERROR: lock already active	Another user is trying to lock NETCONF while it is currently locked.
NETCONF-ERROR: lock time 1 expired closing session 646B7038	A locked NETCONF session has been idle longer than the time configured by the **netconf lock-time** command. The locked NETCONF session is closed.

Related Commands

Command	Description
clear netconf	Clears NETCONF statistics counters, NETCONF sessions, and frees associated resources and locks.
debug cns xml	Turns on debugging messages related to the CNS XML parser.
netconf lock-time	Specifies the maximum time a NETCONF configuration lock is in place without an intermediate operation.
netconf max-sessions	Specifies the maximum number of concurrent NETCONF sessions allowed.
netconf ssh	Enables NETCONF over SSHv2.
show netconf	Displays NETCONF statistics counters and session information.

netconf lock-time

To specify the maximum time a network configuration protocol (NETCONF) configuration lock is in place without an intermediate operation, use the **netconf lock-time** command in global configuration mode. To set the NETCONF configuration lock time to the default value, use the **no** form of this command.

netconf lock-time *seconds*

no netconf lock-time

Syntax Description

seconds	Maximum NETCONF session time in seconds. The valid range is 1 to 300 seconds. The default is 10 seconds.

Command Default

The maximum lock time for a NETCONF session is 10 seconds.

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SRA	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines

NETCONF enables you to set a configuration lock. Setting a configuration lock allows you to have exclusive rights to the configuration in order to apply configuration changes. Other users will not have access to the console during the lock time. If the user who has enabled the configuration lock is inactive, the lock timer expires and the session is ejected, preventing the configuration from being locked out if the user loses network connectivity while they have the configuration locked.

Examples

The following example shows how to limit a NETCONF configuration lock to 60 seconds:

```
netconf lock-time 60
```

Related Commands

Command	Description
clear netconf	Clears NETCONF statistics counters, NETCONF sessions, and frees associated resources and locks.
debug netconf	Enables debugging of NETCONF sessions.
netconf max-sessions	Specifies the maximum number of concurrent NETCONF sessions allowed.
netconf ssh	Enables NETCONF over SSHv2.
show netconf	Displays NETCONF statistics counters and session information.

netconf max-sessions

To specify the maximum number of concurrent network configuration protocol (NETCONF) sessions allowed, use the **netconf max-sessions** command in global configuration mode. To reset the number of concurrent NETCONF sessions allowed to the default value of four sessions, use the **no** form of this command.

netconf max-sessions *session*

no netconf max-sessions

Syntax Description

session	Specifies the total number of concurrent NETCONF sessions allowed. The default is 4. The range is 4 to 16.

Command Default

4 concurrent NETCONF sessions are allowed.

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SRA	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines

You can have multiple NETCONF Network Managers concurrently connected. The **netconf max-sessions** command allows the maximum number of concurrent NETCONF sessions. The number of NETCONF sessions is also limited by the amount of available of vty line configured.

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

Extra NETCONF sessions beyond the maximum are not accepted.

Examples

The following example allows a maximum of 5 concurrent NETCONF sessions:

```
netconf max-sessions 5
```

Related Commands

Command	Description
clear netconf	Clears NETCONF statistics counters, NETCONF sessions, and frees associated resources and locks.
debug netconf	Enables debugging of NETCONF sessions.
netconf lock-time	Specifies the maximum time a NETCONF configuration lock is in place without an intermediate operation.

Command	Description
netconf ssh	Enables NETCONF over SSHv2.
show netconf	Displays NETCONF statistics counters and session information.

netconf ssh

To enable Network Configuration Protocol (NETCONF) over Secure Shell Version 2 (SSHv2), use the **netconf ssh** command in global configuration mode. To disable NETCONF over SSHv2, use the **no** form of this command.

netconf ssh [**acl** *access-list-number*]

no netconf ssh

Syntax Description

acl	(Optional) Specifies an access list to use during NETCONF sessions.
access-list-number	Number of the access-list to use during NETCONF sessions.

Command Default

NETCONF over SSHv2 is not enabled.

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SRA	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines

NETCONF is supported only on SSHv2.

Examples

The following example shows how to enable NETCONF over SSHv2 and apply access-list 1 to NETCONF sessions:

```
netconf ssh acl 1
```

Related Commands

Command	Description
clear netconf	Clears NETCONF statistics counters, NETCONF sessions, and frees associated resources and locks.
debug netconf	Enables debugging of NETCONF sessions.
netconf lock-time	Specifies the maximum time a NETCONF configuration lock is in place without an intermediate operation.
netconf max-sessions	Specifies the maximum number of concurrent NETCONF sessions allowed.
show netconf	Displays NETCONF statistics counters and session information.

show netconf

To display network configuration protocol (NETCONF) statistics counters and session information, use the **show netconf** command in privileged EXEC mode.

show netconf {**counters** | **session**}

Syntax Description

counters	Displays NETCONF statistics and informational counters.
session	Displays the current state of all connected NETCONF sessions across all transports and any resources and locks in use by the session.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRA	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Examples

The following example provides sample output for the **show netconf counters** command:

```
Router# show netconf counters

NETCONF Counters
Connection Attempts:0: rejected:0 no-hello:0 success:0
Transactions
        total:0, success:0, errors:0
detailed errors:
        in-use 0        invalid-value 0         too-big 0
        missing-attribute 0     bad-attribute 0         unknown-attribute 0
        missing-element 0       bad-element 0   unknown-element 0
        unknown-namespace 0     access-denied 0         lock-denied 0
        resource-denied 0       rollback-failed 0       data-exists 0
        data-missing 0  operation-not-supported 0       operation-failed 0
        partial-operation 0
```

The following example provides sample output for the **show netconf session** command:

```
Router# show netconf session

(Current | max) sessions:   3 | 4
Operations received: 100                Operation errors: 99
Connection Requests: 5                  Authentication errors: 2   Connection Failures: 0
ACL dropped : 30
Notifications  Sent: 20
```

Table 2 describes the significant fields shown in the display.

Table 2 show netconf Field Descriptions

Field	Description
Connection Attempts	Number of NETCONF Connection attempts.
rejected	Number of rejected NETCONF sessions.
no-hello	Number of NETCONF sessions that were dropped because Hello messages were not received.
success	Number of successful NETCONF sessions.
in-use 0	The request requires a resource that is already in use.
invalid-value 0	The request specifies an invalid value for one or more parameters.
too-big 0	The request or response that would be generated would be too large for the implementation to handle.
missing-attribute 0	An expected attribute is missing.
bad-attribute 0	An attribute value is incorrect. An attribute that is the incorrect type, out of range, or contains a pattern mismatch will be displayed as a bad attribute.
unknown-attribute 0	An unexpected attribute is present.
missing-element 0	An expected element is missing.
bad-element 0	An element value is not correct. An element that is the incorrect type, out of range, or contains a pattern mismatch will be displayed as a bad element.
unknown-element 0	An unexpected element is present.
unknown-namespace 0	An unexpected name space is present.
access-denied 0	Access to a requested NETCONF session is denied because authorization failed.
lock-denied 0	Access to a requested lock is denied because the lock is currently in use.
resource-denied 0	A request could not be completed because of insufficient resources.
rollback-failed 0	A request to roll back a configuration change was not completed.
data-exists 0	A request could not be completed because the relevant content already exists.
data-missing 0	A request could not be completed because the relevant content does not exist.
operation-not-supported 0	A request could not be completed because the requested operation is not supported.
operation-failed 0	A request could not be completed because the requested operation failed for a reason not specified by another error notice.
partial-operation 0	Part of a requested operation failed or was not attempted.

Table 2 show netconf Field Descriptions (continued)

Field	Description
(Current | max) sessions: 3 | 4	Number of current NETCONF sessions and the maximum number of concurrent NETCONF sessions allowed.
Operations received: 100	Number of NETCONF operations received.
Operation errors: 99	Number of NETCONF operation errors.
Connection Requests: 5	Number of NETCONF connection requests.
Authentication errors: 2	Number of NETCONF authentication errors.
Connection Failures: 0	Number of unsuccessful NETCONF session connections.
ACL dropped: 30	Number of NETCONF sessions dropped due to an access list.
Notifications Sent: 20	Number of NETCONF notifications sent.

Related Commands

Command	Description
clear netconf	Clears NETCONF statistics counters, NETCONF sessions, and frees associated resources and locks.
debug netconf	Enables debugging of NETCONF sessions.
netconf lock-time	Specifies the maximum time a NETCONF configuration lock is in place without an intermediate operation.
netconf max-sessions	Specifies the maximum number of concurrent NETCONF sessions allowed.
netconf ssh	Enables NETCONF over SSHv2.

Feature Information for NETCONF over SSHv2

Table 3 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 3 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 3 Feature Information for NETCONF over SSHv2

Feature Name	Releases	Feature Information
NETCONF over SSHv2	12.2(33)SRA 12.4(9)T	The NETCONF over SSHv2 feature enables you to perform network configurations via Cisco command-line interface (CLI) over an encrypted transport. The NETCONF protocol defines a simple mechanism through which a network device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated. NETCONF uses an Extensible Markup Language (XML)-based data encoding for the configuration data and protocol messages. • In 12.4(9)T, this feature was introduced.