CONTENTS

PREFACE

Preface xi
Changes to This Document xi
Communications, Services, and Additional Information xii

CHAPTER 1

Authentication, Authorization, and Accounting Commands 1

aaa accounting 4
aaa accounting system default 6
aaa accounting update 8
aaa authentication (XR-VM) 9
aaa authorization (XR-VM) 11
aaa default-taskgroup 14
aaa group server radius 15
aaa group server tacacs+ 17
aaa password-policy 18
accounting (line) 21
authorization (line) 22
deadtime (server-group configuration) 23
description (AAA) 24
group (AAA) 25
inherit taskgroup 27
inherit usergroup 28
key (TACACS+) 29
login authentication 30
password (AAA) 32
policy (AAA) 34
Lawful Intercept Commands 269

lawful-intercept disable 270
The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

Notwithstanding any other warranty herein, all document files and software of these suppliers are provided "as is" with all faults. Cisco and the above-named suppliers disclaim all warranties, expressed or implied, including, without limitation, those of merchantability, fitness for a particular purpose and noninfringement or arising from a course of dealing, usage, or trade practice.

In no event shall Cisco or its suppliers be liable for any indirect, special, consequential, or incidental damages, including, without limitation, lost profits or loss or damage to data arising out of the use or inability to use this manual, even if Cisco or its suppliers have been advised of the possibility of such damages.

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

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2015–2020 Cisco Systems, Inc. All rights reserved.
Preface

This preface contains these sections:

- Changes to This Document, on page xi
- Communications, Services, and Additional Information, on page xii

Changes to This Document

This table lists the technical changes made to this document since it was first released.

Table 1: Changes to This Document

<table>
<thead>
<tr>
<th>Date</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 2015</td>
<td>Initial release of this document.</td>
</tr>
<tr>
<td>July 2016</td>
<td>Republished for Release 6.0.2.</td>
</tr>
<tr>
<td>August 2016</td>
<td>Republished for Release 6.1.2.</td>
</tr>
<tr>
<td>February 2017</td>
<td>Republished for Release 6.1.3.</td>
</tr>
<tr>
<td>July 2017</td>
<td>Republished for Release 6.2.2.</td>
</tr>
<tr>
<td>September 2017</td>
<td>Republished for Release 6.3.1.</td>
</tr>
<tr>
<td>March 2018</td>
<td>Republished for Release 6.3.2.</td>
</tr>
<tr>
<td>March 2018</td>
<td>Republished for Release 6.4.1.</td>
</tr>
<tr>
<td>July 2018</td>
<td>Republished for Release 6.4.2.</td>
</tr>
<tr>
<td>August 2018</td>
<td>Republished for Release 6.5.1.</td>
</tr>
<tr>
<td>January 2019</td>
<td>Updated Management Plane Protection Commands chapter for Release 6.5.2</td>
</tr>
</tbody>
</table>
Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you’re looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.
Authentication, Authorization, and Accounting Commands

This module describes the commands used to configure authentication, authorization, and accounting (AAA) services.

Note

All commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router that is introduced from Cisco IOS XR Release 6.3.2. References to earlier releases in Command History tables apply to only the Cisco NCS 5500 Series Router.

Note

- Starting with Cisco IOS XR Release 6.6.25, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 560 Series Routers.
- Starting with Cisco IOS XR Release 6.3.2, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router.
- References to releases before Cisco IOS XR Release 6.3.2 apply to only the Cisco NCS 5500 Series Router.
- Cisco IOS XR Software Release 7.0.1 specific updates are not applicable for the following variants of Cisco NCS 540 Series Routers:
  - N540-28Z4C-SYS-A
  - N540-28Z4C-SYS-D
  - N540X-16Z4G8Q2C-A
  - N540X-16Z4G8Q2C-D
  - N540-12Z20G-SYS-A
  - N540-12Z20G-SYS-D
  - N540X-12Z16G-SYS-A
  - N540X-12Z16G-SYS-D
For detailed information about AAA concepts, configuration tasks, and examples, see the Configuring AAA Services chapter in the System Security Configuration Guide for Cisco NCS 5500 Series Routers.

Currently, only default VRF is supported. VPNv4, VPNv6 and VPN routing and forwarding (VRF) address families will be supported in a future release.

- aaa accounting, on page 4
- aaa accounting system default, on page 6
- aaa accounting update, on page 8
- aaa authentication (XR-VM), on page 9
- aaa authorization (XR-VM), on page 11
- aaa default-taskgroup, on page 14
- aaa group server radius, on page 15
- aaa group server tacacs+, on page 17
- aaa password-policy, on page 18
- accounting (line), on page 21
- authorization (line), on page 22
- deadtime (server-group configuration), on page 23
- description (AAA), on page 24
- group (AAA), on page 25
- inherit taskgroup, on page 27
- inherit usergroup, on page 28
- key (TACACS+), on page 29
- login authentication, on page 30
- password (AAA), on page 32
- policy (AAA), on page 34
- radius-server dead-criteria time, on page 35
- radius-server dead-criteria tries, on page 36
- radius-server deadtime (BNG), on page 37
- radius-server key (BNG), on page 38
- radius-server retransmit (BNG), on page 39
- radius-server timeout (BNG), on page 40
- radius source-interface (BNG), on page 41
- secret, on page 42
- server (RADIUS), on page 45
- server (TACACS+), on page 47
- server-private (RADIUS), on page 48
- server-private (TACACS+), on page 50
- show aaa (XR-VM), on page 52
- show aaa accounting, on page 57
- show aaa password-policy, on page 59
- show radius, on page 61
- show radius accounting, on page 63
- show radius authentication, on page 65
- show radius dead-criteria, on page 67
• show radius server-groups, on page 69
• show tacacs, on page 71
• show tacacs server-groups, on page 73
• show user, on page 74
• show aaa user-group, on page 78
• show tech-support aaa, on page 79
• single-connection, on page 80
• tacacs-server host, on page 81
• tacacs-server key, on page 83
• tacacs-server timeout, on page 84
• tacacs-server ipv4, on page 85
• tacacs source-interface, on page 87
• task, on page 89
• taskgroup, on page 90
• timeout (TACACS+), on page 92
• timeout login response, on page 93
• usergroup, on page 94
• username, on page 95
• users group, on page 100
• vrf (RADIUS), on page 102
• vrf (TACACS+), on page 103
aaa accounting

To create a method list for accounting, use the `aaa accounting` command in the XR EXEC mode. To remove a list name from the system, use the `no` form of this command.

```
aaa accounting {commands | exec | mobile | network | subscriber | system} {default | list-name} {start-stop | stop-only} {none | method}
no aaa accounting {commands | exec | mobile | network} {default | list-name}
```

**Syntax Description**

- `commands` Enables accounting for XR EXEC shell commands.
- `exec` Enables accounting of a XR EXEC session.
- `mobile` Enables Mobile IP related accounting events.
- `network` Enables accounting for all network-related service requests, such as Internet Key Exchange (IKE) and Point-to-Point Protocol (PPP).
- `subscriber` Sets accounting lists for subscribers.
- `system` Enables accounting for all system-related events.
- `event manager` Sets the authorization list for XR EXEC.
- `default` Uses the listed accounting methods that follow this keyword as the default list of methods for accounting services.
- `list-name` Character string used to name the accounting method list.
- `start-stop` Sends a “start accounting” notice at the beginning of a process and a “stop accounting” notice at the end of a process. The requested user process begins regardless of whether the “start accounting” notice was received by the accounting server.
- `stop-only` Sends a “stop accounting” notice at the end of the requested user process.
  
  Note: This is not supported with system accounting.
- `none` Uses no accounting.
- `method` Method used to enable AAA system accounting. The value is one of the following options:
  
  - `group tacacs+`—Uses the list of all TACACS+ servers for accounting.
  - `group radius`—Uses the list of all RADIUS servers for accounting.
  - `group named-group`—Uses a named subset of TACACS+ or RADIUS servers for accounting, as defined by the `aaa group server tacacs+` or `aaa group server radius` command.

**Command Default**

AAA accounting is disabled.

**Command Modes**

XR EXEC mode
Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines

Use the `aaa accounting` command to create default or named method lists defining specific accounting methods and that can be used on a per-line or per-interface basis. You can specify up to four methods in the method list. The list name can be applied to a line (console, aux, or vty template) to enable accounting on that particular line.

The Cisco IOS XR software supports both TACACS+ and RADIUS methods for accounting. The router reports user activity to the security server in the form of accounting records, which are stored on the security server.

Method lists for accounting define the way accounting is performed, enabling you to designate a particular security protocol that is used on specific lines or interfaces for particular types of accounting services.

For minimal accounting, include the `stop-only` keyword to send a “stop accounting” notice after the requested user process. For more accounting, you can include the `start-stop` keyword, so that TACACS+ or RADIUS sends a “start accounting” notice at the beginning of the requested process and a “stop accounting” notice after the process. The accounting record is stored only on the TACACS+ or RADIUS server.

The requested user process begins regardless of whether the “start accounting” notice was received by the accounting server.

Note

This command cannot be used with TACACS or extended TACACS.

Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

Examples

The following example shows how to define a default commands accounting method list, where accounting services are provided by a TACACS+ security server, with a stop-only restriction:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa accounting commands default stop-only group tacacs+
```
aaa accounting system default

To enable authentication, authorization, and accounting (AAA) system accounting, use the `aaa accounting system default` command in the XR Config mode. To disable system accounting, use the `no` form of this command.

```
aaa accounting system default {start-stop | stop-only} {none | method}
no aaa accounting system default
```

**Syntax Description**

- `start-stop` Sends a “start accounting” notice during system bootup and a “stop accounting” notice during system shutdown or reload.
- `stop-only` Sends a “stop accounting” notice during system shutdown or reload.
- `none` Uses no accounting.
- `method` Method used to enable AAA system accounting. The value is one of the following options:
  - `group tacacs+`—Uses the list of all TACACS+ servers for accounting.
  - `group radius`—Uses the list of all RADIUS servers for accounting.
  - `group named-group`—Uses a named subset of TACACS+ or RADIUS servers for accounting, as defined by the `aaa group server tacacs+` or `aaa group server radius` command.

**Command Default**

AAA accounting is disabled.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

System accounting does not use named accounting lists; you can define only the default list for system accounting.

The default method list is automatically applied to all interfaces or lines. If no default method list is defined, then no accounting takes place.

You can specify up to four methods in the method list.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

This example shows how to cause a “start accounting” record to be sent to a TACACS+ server when a router initially boots. A “stop accounting” record is also sent when a router is shut down or reloaded.
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa accounting system default start-stop group tacacs+
aaa accounting update

To enable periodic interim accounting records to be sent to the accounting server, use the `aaa accounting update` command in the XR Config mode. To disable the interim accounting updates, use the `no` form of this command.

`aaa accounting update {periodic minutes}`
`no aaa accounting update`

**Syntax Description**

- `periodic minutes` (Optional) Sends an interim accounting record to the accounting server periodically, as defined by the `minutes` argument, which is an integer that specifies the number of minutes. The range is from 1 to 35791394 minutes.

**Command Default**

AAA accounting update is disabled.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

When used with the `periodic` keyword, interim accounting records are sent periodically as defined by the `minutes` argument. The interim accounting record contains all the accounting information recorded for that user up to the time the accounting record is sent.

⚠️ **Caution**

Using the `aaa accounting update` command with the `periodic` keyword can cause heavy congestion when many users are logged into the network.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to send periodic interim accounting records to the RADIUS server at 30-minute intervals:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa accounting update periodic 30
```
aaa authentication (XR-VM)

To create a method list for authentication, use the `aaa authentication` command in the XR Config mode or System Admin Config mode. To disable this authentication method, use the `no` form of this command.

```
aaa authentication {<login | ppp> } {<default|list-name> } method-list
no aaa authentication {<login | ppp> } {<default|list-name> } method-list
```

**Syntax Description**

- `login` Sets authentication for login.
- `ppp` Sets authentication for Point-to-Point Protocol.
- `default` Uses the listed authentication methods that follow this keyword as the default list of methods for authentication.
- `subscriber` Sets the authentication list for the subscriber.
- `list-name` Character string used to name the authentication method list.
- `method-list` Method used to enable AAA system accounting. The value is one of the following options:
  - `group tacacs+`—Specifies a method list that uses the list of all configured TACACS+ servers for authentication.
  - `group radius`—Specifies a method list that uses the list of all configured RADIUS servers for authentication.
  - `group named-group`—Specifies a method list that uses a named subset of TACACS+ or RADIUS servers for authentication, as defined by the `aaa group server tacacs+` or `aaa group server radius` command.
  - `local`—Specifies a method list that uses the local username database method for authentication. AAA method rollover happens beyond the local method if username is not defined in the local group.
  - `line`—Specifies a method list that uses the line password for authentication.

**Command Default**

Default behavior applies the local authentication on all ports.

**Command Modes**

XR Config mode or System Admin Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `aaa authentication` command to create a series of authentication methods, or method list. You can specify up to four methods in the method list. A `method list` is a named list describing the authentication methods (such as TACACS+ or RADIUS) in sequence. The subsequent methods of authentication are used only if the initial method is not available, not if it fails.

The default method list is applied for all interfaces for authentication, except when a different named method list is explicitly specified—in which case the explicitly specified method list overrides the default list.

For console and vty access, if no authentication is configured, a default of local method is applied.
The `group tacacs+`, `group radius`, and `group group-name` forms of this command refer to a set of previously defined TACACS+ or RADIUS servers.

- Use the `tacacs-server host` or `radius-server host` command to configure the host servers.
- Use the `aaa group server tacacs+` or `aaa group server radius` command to create a named subset of servers.
- The `login` keyword, `local` option, and `group` option are available only in System Admin Config mode.

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read,</td>
<td>write</td>
</tr>
</tbody>
</table>

The following example shows how to specify the default method list for authentication, and also enable authentication for console in XR Config mode:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa authentication login default group tacacs+
```
## aaa authorization (XR-VM)

To create a method list for authorization, use the `aaa authorization` command in the XR Config mode. To disable authorization for a function, use the `no` form of this command.

```plaintext
aaa authorization {commands | eventmanager | exec | network | subscriber} {default list-name} {none | local | group {tacacs+ | radius group-name}}
no aaa authorization {commands | eventmanager | exec | network | subscriber} {default list-name}
```

### Syntax Description

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>commands</td>
<td>Configures authorization for all XR EXEC mode shell commands.</td>
</tr>
<tr>
<td>eventmanager</td>
<td>Applies an authorization method for authorizing an event manager (fault manager).</td>
</tr>
<tr>
<td>exec</td>
<td>Configures authorization for an interactive (XR EXEC mode) session.</td>
</tr>
<tr>
<td>network</td>
<td>Configures authorization for network services, such as PPP or Internet Key Exchange (IKE).</td>
</tr>
<tr>
<td>subscriber</td>
<td>Sets the authorization lists for the subscriber.</td>
</tr>
<tr>
<td>default</td>
<td>Uses the listed authorization methods that follow this keyword as the default list of methods for authorization.</td>
</tr>
<tr>
<td>list-name</td>
<td>Character string used to name the list of authorization methods.</td>
</tr>
<tr>
<td>none</td>
<td>Uses no authorization. If you specify <code>none</code>, no subsequent authorization methods is attempted. However, the task ID authorization is always required and cannot be disabled.</td>
</tr>
<tr>
<td>local</td>
<td>Uses local authorization. This method of authorization is not available for command authorization.</td>
</tr>
<tr>
<td>group tacacs+</td>
<td>Uses the list of all configured TACACS+ servers for authorization.</td>
</tr>
<tr>
<td>group radius</td>
<td>Uses the list of all configured RADIUS servers for authorization. This method of authorization is not available for command authorization.</td>
</tr>
<tr>
<td>group group-name</td>
<td>Uses a named subset of TACACS+ or RADIUS servers for authorization as defined by the <code>aaa group server tacacs+</code> or <code>aaa group server radius</code> command.</td>
</tr>
</tbody>
</table>

### Command Default

Authorization is disabled for all actions (equivalent to the method `none` keyword).

### Command Modes

XR Config mode

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

Use the `aaa authorization` command to create method lists defining specific authorization methods that can be used on a per-line or per-interface basis. You can specify up to four methods in the method list.
The command authorization mentioned here applies to the one performed by an external AAA server and not for task-based authorization.

Method lists for authorization define the ways authorization will be performed and the sequence in which these methods will be performed. A method list is a named list describing the authorization methods (such as TACACS+), in sequence. Method lists enable you to designate one or more security protocols for authorization, thus ensuring a backup system in case the initial method fails. Cisco IOS XR software uses the first method listed to authorize users for specific network services; if that method fails to respond, Cisco IOS XR software selects the next method listed in the method list. This process continues until there is successful communication with a listed authorization method or until all methods defined have been exhausted.

Cisco IOS XR software attempts authorization with the next listed method only when there is no response (not a failure) from the previous method. If authorization fails at any point in this cycle—meaning that the security server or local username database responds by denying the user services—the authorization process stops and no other authorization methods are attempted.

The Cisco IOS XR software supports the following methods for authorization:

- **none**—The router does not request authorization information; authorization is not performed over this line or interface.
- **local**—Use the local database for authorization.
- **group tacacs+**—Use the list of all configured TACACS+ servers for authorization.
- **group radius**—Use the list of all configured RADIUS servers for authorization.
- **group group-name**—Uses a named subset of TACACS+ or RADIUS servers for authorization.

Method lists are specific to the type of authorization being requested. Cisco IOS XR software supports four types of AAA authorization:

- **Commands authorization**—Applies to the XR EXEC mode commands a user issues. Command authorization attempts authorization for all XR EXEC mode commands.

  **Note**
  
  “Command” authorization is distinct from “task-based” authorization, which is based on the task profile established during authentication.

- **XR EXEC mode authorization**—Applies authorization for starting an XR EXEC mode session.

  **Note**
  
  The `exec` keyword is no longer used to authorize the fault manager service. The `eventmanager` keyword (fault manager) is used to authorize the fault manager service. The `exec` keyword is used for EXEC authorization.

- **Network authorization**—Applies authorization for network services, such as IKE.
- **Event manager authorization**—Applies an authorization method for authorizing an event manager (fault manager). You are allowed to use TACACS+ or locald.
The `eventmanager` keyword (fault manager) replaces the `exec` keyword to authorize event managers (fault managers).

When you create a named method list, you are defining a particular list of authorization methods for the indicated authorization type. When defined, method lists must be applied to specific lines or interfaces before any of the defined methods are performed.

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to define the network authorization method list named listname1, which specifies that TACACS+ authorization is used:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa authorization commands listname1 group tacacs+
```
aaa default-taskgroup

To specify a task group for both remote TACACS+ authentication and RADIUS authentication, use the `aaa default-taskgroup` command in the XR Config mode. To remove this default task group, enter the `no` form of this command.

```
aaa default-taskgroup taskgroup-name
no aaa default-taskgroup
```

### Syntax Description

- `taskgroup-name` Name of an existing task group.

### Command Default

No default task group is assigned for remote authentication.

### Command Modes

XR Config mode

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

Use the `aaa default-taskgroup` command to specify an existing task group for remote TACACS+ authentication.

### Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

### Examples

The following example shows how to specify taskgroup1 as the default task group for remote TACACS+ authentication:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa default-taskgroup taskgroup1
```
**aaa group server radius**

To group different RADIUS server hosts into distinct lists, use the `aaa group server radius` command in the XR Config mode. To remove a group server from the configuration list, enter the `no` form of this command.

```
aaa group server radius  group-name
no aaa group server radius  group-name
```

**Syntax Description**

- `group-name` Character string used to name the group of servers.

**Command Default**

This command is not enabled.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `aaa group server radius` command to group existing server hosts, which allows you to select a subset of the configured server hosts and use them for a particular service. A server group is used in conjunction with a global server-host list. The server group lists the IP addresses or hostnames of the selected server hosts.

Server groups can also include multiple host entries for the same server, as long as each entry has a unique identifier. The combination of an IP address and User Datagram Protocol (UDP) port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific authentication, authorization, and accounting (AAA) service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service, for example, accounting, the second host entry acts as an automatic switchover backup to the first host entry. Using this example, if the first host entry fails to provide accounting services, the network access server tries the second host entry on the same device for accounting services. The RADIUS host entries are tried in the order in which they are configured in the server group.

All members of a server group must be the same type, that is, RADIUS.

The server group cannot be named radius or tacacs.

This command enters server group configuration mode. You can use the server command to associate a particular RADIUS server with the defined server group.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows the configuration of an AAA group server named radgroup1, which comprises three member servers:
aaa group server radius

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa group server radius radgroup1
RP/0/RP0/CPU0:router(config-sg-radius)# server 10.0.0.5 auth-port 1700 acct-port 1701
RP/0/RP0/CPU0:router(config-sg-radius)# server 10.0.0.10 auth-port 1702 acct-port 1703
RP/0/RP0/CPU0:router(config-sg-radius)# server 10.0.0.20 auth-port 1705 acct-port 1706

Note

If the auth-port port-number and acct-port port-number keywords and arguments are not specified, the default value of the port-number argument for the auth-port keyword is 1645 and the default value of the port-number argument for the acct-port keyword is 1646.
aaa group server tacacs+

To group different TACACS+ server hosts into distinct lists, use the `aaa group server tacacs+` command in the XR Config mode. To remove a server group from the configuration list, enter the `no` form of this command.

```
aaa group server tacacs+  group-name
no aaa group server tacacs+  group-name
```

**Syntax Description**

- `group-name` Character string used to name a group of servers.

**Command Default**

This command is not enabled.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The AAA server-group feature introduces a way to group existing server hosts. The feature enables you to select a subset of the configured server hosts and use them for a particular service.

The `aaa group server tacacs+` command enters server group configuration mode. The `server` command associates a particular TACACS+ server with the defined server group.

A server group is a list of server hosts of a particular type. The supported server host type is TACACS+ server hosts. A server group is used with a global server host list. The server group lists the IP addresses or hostnames of the selected server hosts.

The server group cannot be named radius or tacacs.

**Note**

Group name methods refer to a set of previously defined TACACS+ servers. Use the `tacacs-server host` command to configure the host servers.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows the configuration of an AAA group server named tacgroup1, which comprises three member servers:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa group server tacacs+ tacgroup1
RP/0/RP0/CPU0:router(config-sg-tacacs)# server 192.168.200.226
RP/0/RP0/CPU0:router(config-sg-tacacs)# server 192.168.200.227
RP/0/RP0/CPU0:router(config-sg-tacacs)# server 192.168.200.228
```
aaa password-policy

To define a AAA password security policy, use the `aaa password-policy` command in XR Config mode. To remove the AAA password security policy, use the `no` form of this command.

```
aaa password-policy policy-name {min-length min-length | max-length max-length | special-char special-char | upper-case upper-case | lower-case lower-case | numeric numeric | lifetime {years | months | days | hours | minutes | seconds} lifetime | min-char-change min-char-change} authen-max-attempts authen-max-attempts | lockout-time {days | hours | minutes | seconds} lockout-time
```

### Syntax Description

- **policy-name**: Specifies the name of the password, in characters.
- **min-length**: Specifies the minimum length of the password, in integer.
- **max-length**: Specifies the maximum length of the password, in integer.
- **special-char**: Specifies the number of special characters allowed in the password policy, in integer.
- **upper-case**: Specifies the number of upper case alphabets allowed in the password policy, in integer.
- **lower-case**: Specifies the number of lower case alphabets allowed in the password policy, in integer.
- **numeric**: Specifies the number of numerals allowed in the password policy, in integer.
- **lifetime**: Specifies the maximum lifetime for the password, the value of which is specified in integer, as years, months, days, hours, minutes or seconds.
- **min-char-change**: Specifies the number of character change required between subsequent passwords, in integer.
- **authen-max-attempts**: Specifies, in integer, the maximum number of authentication failure attempts allowed for a user, in order to restrict users who authenticate with invalid login credentials.
- **lockout-time**: Specifies, in integer, the duration (in days, hours, minutes or seconds) for which the user is locked out when he exceeds the maximum limit of authentication failure attempts allowed.

### Command Default

None

### Command Modes

XR Config mode

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.2.1</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>Release 7.2.1</td>
<td>The command options (except a few mentioned in the usage guidelines section) were extended to user secret as well.</td>
</tr>
</tbody>
</table>
**Usage Guidelines**

AAA password security policy works as such for Cisco IOS XR platforms. Whereas, this feature is supported only on XR VM, for Cisco IOS XR 64 bit platforms and Cisco NCS 5500 Series Routers.

For more details on the usage of each option of this command, refer the section on **AAA Password Security for FIPS Compliance** in Configuring AAA Services chapter in the System Security Configuration Guide for Cisco NCS 5500 Series Routers.

You must configure both authen-max-attempts and lockout-time in order for the lock out functionality to take effect.

The min-char-change option is effective only for password change through logon, and not for password change by configuration.

Use **username** command along with **password-policy** option, in the XR Config mode, to associate the password policy with a particular user.

From Cisco IOS XR Software Release 7.2.1 and later, most of the options of the **aaa password-policy** command listed in the syntax above are applicable to user password as well as secret. Whereas, the options listed below are supported only for password, and not for secret:

- max-char-repetition
- min-char-change
- restrict-password-reverse
- restrict-password-advanced

This table lists the default, maximum and minimum values of various command variables:

<table>
<thead>
<tr>
<th>Command Variables</th>
<th>Default Value</th>
<th>Maximum Value</th>
<th>Minimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>policy-name</strong></td>
<td>None</td>
<td>253</td>
<td>1</td>
</tr>
<tr>
<td><strong>max-length</strong></td>
<td>253</td>
<td>253</td>
<td>2</td>
</tr>
<tr>
<td><strong>min-length</strong></td>
<td>2</td>
<td>253</td>
<td>2</td>
</tr>
<tr>
<td><strong>special-char</strong></td>
<td>0</td>
<td>253</td>
<td>0</td>
</tr>
<tr>
<td><strong>upper-case</strong></td>
<td>0</td>
<td>253</td>
<td>0</td>
</tr>
<tr>
<td><strong>lower-case</strong></td>
<td>0</td>
<td>253</td>
<td>0</td>
</tr>
<tr>
<td><strong>numeric</strong></td>
<td>0</td>
<td>253</td>
<td>0</td>
</tr>
<tr>
<td>For <strong>lifetime</strong>:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>years</td>
<td>0</td>
<td>99</td>
<td>1</td>
</tr>
<tr>
<td>months</td>
<td>0</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>days</td>
<td>0</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>hours</td>
<td>0</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>minutes</td>
<td>0</td>
<td>59</td>
<td>1</td>
</tr>
<tr>
<td>seconds</td>
<td>0</td>
<td>59</td>
<td>1</td>
</tr>
</tbody>
</table>
### Command Variables

<table>
<thead>
<tr>
<th>Command Variables</th>
<th>Default Value</th>
<th>Maximum Value</th>
<th>Minimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>min-char-change</td>
<td>4</td>
<td>253</td>
<td>0</td>
</tr>
<tr>
<td>authen-max-attempts</td>
<td>0</td>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td>For lockout-time :</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>days</td>
<td>0</td>
<td>255</td>
<td>1</td>
</tr>
<tr>
<td>hours</td>
<td>0</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>minutes</td>
<td>0</td>
<td>59</td>
<td>1</td>
</tr>
<tr>
<td>seconds</td>
<td>0</td>
<td>59</td>
<td>1</td>
</tr>
</tbody>
</table>

This example shows how to define a AAA password security policy:

```
RP/0/RP0/CPU0:router(config)#aaa password-policy test-policy
RP/0/RP0/CPU0:router(config-aaa)#min-length 8
RP/0/RP0/CPU0:router(config-aaa)#max-length 15
RP/0/RP0/CPU0:router(config-aaa)#lifetime months 3
RP/0/RP0/CPU0:router(config-aaa)#min-char-change 5
RP/0/RP0/CPU0:router(config-aaa)#authen-max-attempts 3
RP/0/RP0/CPU0:router(config-aaa)#lockout-time days 1
```

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show aaa password-policy</td>
<td>Displays the details of AAA password policy.</td>
</tr>
</tbody>
</table>

username, on page 95
accounting (line)

To enable authentication, authorization, and accounting (AAA) accounting services for a specific line or group of lines, use the `accounting` command. To disable AAA accounting services, use the `no` form of this command.

```
accounting {commands | exec} {default|list-name}
no accounting {commands | exec}
```

**Syntax Description**

- `commands` Enables accounting on the selected lines for all XR EXEC mode shell commands.
- `exec` Enables accounting of XR EXEC mode session.
- `default` The name of the default method list, created with the `aaa accounting` command.
- `list-name` Specifies the name of a list of accounting methods to use. The list is created with the `aaa accounting` command.

**Command Default**

Accounting is disabled.

**Command Modes**

Line template configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

After you enable the `aaa accounting` command and define a named accounting method list (or use the default method list) for a particular type of accounting, you must apply the defined lists to the appropriate lines for accounting services to take place. Use the `accounting` command to apply the specified method lists to the selected line or group of lines. If a method list is not specified this way, no accounting is applied to the selected line or group of lines.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to enable command accounting services using the accounting method list named `listname2` on a line template named `configure`:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# line template configure
RP/0/RP0/CPU0:router(config-line)# accounting commands listname2
```
authorization (line)

To enable authentication, authorization, and accounting (AAA) authorization for a specific line or group of
lines, use the `authorization` command in line template configuration mode. To disable authorization, use the `no` form of this command.

```
authorization {commands | exec | eventmanager} {default|list-name}
no authorization {commands | exec | eventmanager}
```

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>commands</code></td>
<td>Enables authorization on the selected lines for all commands.</td>
</tr>
<tr>
<td><code>exec</code></td>
<td>Enables authorization for an interactive XR EXEC mode session.</td>
</tr>
<tr>
<td><code>default</code></td>
<td>Applies the default method list, created with the <code>aaa authorization</code> command.</td>
</tr>
<tr>
<td><code>eventmanager</code></td>
<td>Sets eventmanager authorization method. This method is used for the embedded event manager.</td>
</tr>
<tr>
<td><code>list-name</code></td>
<td>Specifies the name of a list of authorization methods to use. If no list name is specified, the system uses the default. The list is created with the <code>aaa authorization</code> command.</td>
</tr>
</tbody>
</table>

### Command Default

Authorization is not enabled.

### Command Modes

Line template configuration

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

After you use the `aaa authorization` command to define a named authorization method list (or use the default method list) for a particular type of authorization, you must apply the defined lists to the appropriate lines for authorization to take place. Use the `authorization` command to apply the specified method lists (or, if none is specified, the default method list) to the selected line or group of lines.

### Task ID

<table>
<thead>
<tr>
<th>Task</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

### Examples

The following example shows how to enable command authorization using the method list named `listname4` on a line template named `configure`:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# line template configure
RP/0/RP0/CPU0:router(config-line)# authorization commands listname4
```
deadtime (server-group configuration)

To configure the deadtime value at the RADIUS server group level, use the `deadtime` command in server-group configuration mode. To set deadtime to 0, use the `no` form of this command.

```
deadtime minutes
no deadtime
```

**Syntax Description**

- `minutes` Length of time, in minutes, for which a RADIUS server is skipped over by transaction requests, up to a maximum of 1440 (24 hours). The range is from 1 to 1440.

**Command Default**

Deadtime is set to 0.

**Command Modes**

Server-group configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The value of the deadtime set in the server groups overrides the deadtime that is configured globally. If the deadtime is omitted from the server group configuration, the value is inherited from the primary list. If the server group is not configured, the default value of 0 applies to all servers in the group. If the deadtime is set to 0, no servers are marked dead.

**Examples**

The following example specifies a one-minute deadtime for RADIUS server group `group1` when it has failed to respond to authentication requests for the `deadtime` command:

```
RP/0/RP0/CPU0# configure
RP/0/RP0/CPU0(config)# aaa group server radius group1
RP/0/RP0/CPU0(config-sg-radius)# server 1.1.1.1 auth-port 1645 acct-port 1646
RP/0/RP0/CPU0(config-sg-radius)# server 2.2.2.2 auth-port 2000 acct-port 2001
RP/0/RP0/CPU0(config-sg-radius)# deadtime 1
```
description (AAA)

To create a description of a task group or user group during configuration, use the `description` command in task group configuration or user group configuration mode. To delete a task group description or user group description, use the `no` form of this command.

```
description string
no description
```

**Syntax Description**

- `string` Character string describing the task group or user group.

**Command Default**

None

**Command Modes**

- Task group configuration
- User group configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `description` command inside the task or user group configuration submode to define a description for the task or user group, respectively.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows the creation of a task group description:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# taskgroup alpha
RP/0/RP0/CPU0:router(config-tg)# description this is a sample taskgroup
```

The following example shows the creation of a user group description:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# usergroup alpha
RP/0/RP0/CPU0:router(config-ug)# description this is a sample user group
```
group (AAA)

To add a user to a group, use the `group` command in username configuration mode. To remove the user from a group, use the `no` form of this command.

```
group {cisco-support | maintenance | netadmin | operator | provisioning | retrieve | root-lr | serviceadmin | sysadmin} group-name
no group {cisco-support | maintenance | netadmin | operator | provisioning | retrieve | root-lr | serviceadmin | sysadmin} group-name
```

**Syntax Description**

- `cisco-support` Adds the user to the predefined Cisco support personnel group.

**Note** Starting from IOS XR 6.0 release, the cisco-support group is combined with the root-system group. This means a user who is part of the root-system group can also access commands that are included in the cisco-support group.

- `maintenance` Adds the user to the predefined maintenance group.

- `netadmin` Adds the user to the predefined network administrators group.

- `operator` Adds the user to the predefined operator group.

- `provisioning` Adds the user to the predefined provisioning group.

- `retrieve` Adds the user to the predefined retrieve group.

- `root-lr` Adds the user to the predefined root-lr group. Only users with root-lr authority may use this option.

- `serviceadmin` Adds the user to the predefined service administrators group.

- `sysadmin` Adds the user to the predefined system administrators group.

- `group-name` Adds the user to a named user group that has already been defined with the `usergroup` command.

**Command Default**

None

**Command Modes**

Username configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `group` command in username configuration mode. To access username configuration mode, use the `username`, on page 95 command in XR Config mode.

If the `group` command is used in System Admin Config mode, only cisco-support keywords can be specified.

The privileges associated with the cisco-support group are now included in the root-system group. The cisco-support group is no longer required to be used for configuration.
The following example shows how to assign the user group operator to the user named user1:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# username user1
RP/0/RP0/CPU0:router(config-un)# group operator
```
inherit taskgroup

To enable a task group to derive permissions from another task group, use the inherit taskgroup command in task group configuration mode.

```
inherit taskgroup {taskgroup-name | netadmin | operator | sysadmin | cisco-support | root-lr | serviceadmin}
```

**Syntax Description**

- **taskgroup-name** Name of the task group from which permissions are inherited.
- **netadmin** Inherits permissions from the network administrator task group.
- **operator** Inherits permissions from the operator task group.
- **sysadmin** Inherits permissions from the system administrator task group.
- **cisco-support** Inherits permissions from the cisco support task group.
- **root-lr** Inherits permissions from the root-lr task group.
- **serviceadmin** Inherits permissions from the service administrators task group.

**Command Default**

None

**Command Modes**

Task group configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the inherit taskgroup command to inherit the permissions (task IDs) from one task group into another task group. Any changes made to the taskgroup from which they are inherited are reflected immediately in the group from which they are inherited.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

In the following example, the permissions of task group tg2 are inherited by task group tg1:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# taskgroup tg1
RP/0/RP0/CPU0:router(config-tg)# inherit taskgroup tg2
RP/0/RP0/CPU0:router(config-tg)# end
```
**inherit usergroup**

To enable a user group to derive characteristics of another user group, use the `inherit usergroup` command in user group configuration mode.

```
inherit usergroup  usergroup-name
```

**Syntax Description**

- `usergroup-name` Name of the user group from which permissions are to be inherited.

**Command Default**

None

**Command Modes**

User group configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Each user group is associated with a set of task groups applicable to the users in that group. A task group is defined by a collection of task IDs. Task groups contain task ID lists for each class of action. The task permissions for a user are derived (at the start of the EXEC or XML session) from the task groups associated with the user groups to which that user belongs.

User groups support inheritance from other user groups. Use the `inherit usergroup` command to copy permissions (task ID attributes) from one user group to another user group. The “destination” user group inherits the properties of the inherited group and forms a union of all task IDs specified in those groups. For example, when user group A inherits user group B, the task map of the user group A is a union of that of A and B. Cyclic inclusions are detected and rejected. User groups cannot inherit properties from predefined groups, such as root-system users, root-sdr users, netadmin users, and so on. Any changes made to the usergroup from which it is inherited are reflected immediately in the group from which it is inherited.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to enable the purchasing user group to inherit properties from the sales user group:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# usergroup purchasing
RP/0/RP0/CPU0:router(config-ug)# inherit usergroup sales
```
key (TACACS+)

To specify an authentication and encryption key shared between the AAA server and the TACACS+ server, use the `key (TACACS+)` command in TACACS host configuration mode. To disable this feature, use the `no` form of this command.

```
key {0 clear-text-key | 7 encrypted-key auth-key}
no key {0 clear-text-key | 7 encrypted-key auth-key}
```

### Syntax Description

- **0 clear-text-key**
  - Specifies an unencrypted (cleartext) shared key.
- **7 encrypted-key**
  - Specifies an encrypted shared key.
- **auth-key**
  - Specifies the unencrypted key between the AAA server and the TACACS+ server.

### Command Default

None

### Command Modes

TACACS host configuration

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

The TACACS+ packets are encrypted using the key, and it must match the key used by the TACACS+ daemon. Specifying this key overrides the key set by the `tacacs-server key` command for this server only.

The key is used to encrypt the packets that are going from TACACS+, and it should match with the key configured on the external TACACS+ server so that the packets are decrypted properly. If a mismatch occurs, the result fails.

### Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

### Examples

The following example shows how to set the encrypted key to anykey

```
RP/0/RP0/CPU0:router(config)# tacacs-server host 209.165.200.226
RP/0/RP0/CPU0:router(config-tacacs-host)# key anykey
```
**login authentication**

To enable authentication, authorization, and accounting (AAA) authentication for logins, use the `login authentication` command in line template configuration mode. To return to the default authentication settings, use the `no` form of this command.

```
login authentication {default | list-name}
no login authentication
```

**Syntax Description**
- **default**
  - Default list of AAA authentication methods, as set by the `aaa authentication login` command.
- **list-name**
  - Name of the method list used for authenticating. You specify this list with the `aaa authentication login` command.

**Command Default**
This command uses the default set with the `aaa authentication login` command.

**Command Modes**
- Line template configuration

**Command History**
- **Release**
  - **Modification**
  - Release 6.0
    - This command was introduced.

**Usage Guidelines**
The `login authentication` command is a per-line command used with AAA that specifies the name of a list of AAA authentication methods to try at login.

⚠️ **Caution**
If you use a `list-name` value that was not configured with the `aaa authentication login` command, the configuration is rejected.

Entering the `no` form of the `login authentication` command has the same effect as entering the command with the `default` keyword.

Before issuing this command, create a list of authentication processes by using the `aaa authentication login` command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
<tr>
<td>tty-access</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**
The following example shows that the default AAA authentication is used for the line template `template1`:  

```
RP/0/RP0/CPU0:router# configure
```
RP/0/RP0/CPU0:router(config)# line template template1
RP/0/RP0/CPU0:router(config-line)# login authentication default

The following example shows that the AAA authentication list called list1 is used for the line template template2:

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# line template template2
RP/0/RP0/CPU0:router(config-line)# login authentication list1
password (AAA)

To create a login password for a user, use the `password` command in username configuration mode or line template configuration mode. To remove the password, use the `no` form of this command.

```
password {[0] | 7 password}
no password {0 | 7 password}
```

**Syntax Description**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0</strong></td>
<td>(Optional) Specifies that an unencrypted clear-text password follows.</td>
</tr>
<tr>
<td><strong>7</strong></td>
<td>Specifies that an encrypted password follows.</td>
</tr>
</tbody>
</table>

**password** Specifies the unencrypted password text to be entered by the user to log in, for example, “lab”. If encryption is configured, the password is not visible to the user.

Can be up to 253 characters in length.

**Command Default**
The password is in unencrypted clear text.

**Command Modes**
Username configuration
Line template configuration

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
You can specify one of two types of passwords: encrypted or clear text.

When an XR EXEC mode process is started on a line that has password protection, the process prompts for the password. If the user enters the correct password, the process issues the prompt. The user can try three times to enter a password before the process exits and returns the terminal to the idle state.

Passwords are two-way encrypted and should be used for applications such as PPP that need decryptable passwords that can be decrypted.

**Note**
The `show running-config` command always displays the clear-text login password in encrypted form when the `0` option is used.

**Task ID**
```
<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>
```

**Examples**
The following example shows how to establish the unencrypted password `pwd1` for user. The output from the `show` command displays the password in its encrypted form.
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# username user1
RP/0/RP0/CPU0:router(config-un)# password 0 pwd1
RP/0/RP0/CPU0:router(config-un)# commit
RP/0/RP0/CPU0:router(config-un)# show running-config
Building configuration...
username user1
  password 7 141B1309
To configure a policy that is common for user password as well as secret, use the `policy` command in `username` configuration mode. To remove this configuration, use the `no` form of this command.

```
policy policy-name
```

**Syntax Description**
- `policy-name`: Specifies the name of the policy that is common for user password as well as secret.

**Command Default**
None

**Command Modes**
- `username`

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

For detailed usage guidelines for this command, see the *Guidelines to Configure Password Policy for User Secret* section in the *System Security Configuration Guide for Cisco NCS 5500 Series Routers*.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

This example shows how to configure a password policy that applies to both the password and the secret of the user.

```
Router(config-un)#secret 10
$6$dmwuW0Ajicf98W0.8y/vzyn0Fl/OcGx9d6ds79VwY5Z2EJoHLd7Tic5mOo8ITvr1YGA500A.wJvXFO7IoZry.D88rE3SN2B4zB3e0
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>username</code>, on page 95</td>
<td></td>
</tr>
</tbody>
</table>
**radius-server dead-criteria time**

To specify the minimum amount of time, in seconds, that must elapse from the time that the router last received a valid packet from the RADIUS server to the time the server is marked as dead, use the `radius-server dead-criteria time` command in XR Config mode. To disable the criteria that were set, use the `no` form of this command.

```
radius-server dead-criteria time  seconds
no radius-server dead-criteria time  seconds
```

### Syntax Description

```
seconds  Length of time, in seconds. The range is from 1 to 120 seconds. If the `seconds` argument is not configured, the number of seconds ranges from 10 to 60, depending on the transaction rate of the server.

**Note**  The time criterion must be met for the server to be marked as dead.
```

### Command Default

If this command is not used, the number of seconds ranges from 10 to 60 seconds, depending on the transaction rate of the server.

### Command Modes

XR Config mode

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

If you configure the `radius-server dead-criteria time` command before the `radius-server deadtime` command, the `radius-server dead-criteria time` command may not be enforced.

If a packet has not been received since the router booted and there is a timeout, the time criterion is treated as though it were met.

### Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

### Examples

The following example shows how to establish the time for the dead-criteria conditions for a RADIUS server to be marked as dead for the `radius-server dead-criteria time` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# radius-server dead-criteria time 5
```
radius-server dead-criteria tries

To specify the number of consecutive timeouts that must occur on the router before the RADIUS server is marked as dead, use the `radius-server dead-criteria tries` command in the XR Config mode. To disable the criteria that were set, use the `no` form of this command.

```
radius-server dead-criteria tries
no radius-server dead-criteria tries
```

**Syntax Description**

- `tries` Number of timeouts from 1 to 100. If the `tries` argument is not configured, the number of consecutive timeouts ranges from 10 to 100, depending on the transaction rate of the server and the number of configured retransmissions.

  **Note** The tries criterion must be met for the server to be marked as dead.

**Command Default**

If this command is not used, the number of consecutive timeouts ranges from 10 to 100, depending on the transaction rate of the server and the number of configured retransmissions.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

If the server performs both authentication and accounting, both types of packet are included in the number. Improperly constructed packets are counted as though they were timeouts. All transmissions, including the initial transmit and all retransmits, are counted.

**Note**

If you configure the `radius-server dead-criteria tries` command before the `radius-server deadtime` command, the `radius-server dead-criteria tries` command may not be enforced.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to establish the number of tries for the dead-criteria conditions for a RADIUS server to be marked as dead for the `radius-server dead-criteria tries` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# radius-server dead-criteria tries 4
```
radius-server deadtime (BNG)

To improve RADIUS response times when some servers are unavailable and cause the unavailable servers to be skipped immediately, use the `radius-server deadtime` command in the XR Config mode. To set deadtime to 0, use the `no` form of this command.

```
radius-server deadtime value
no radius-server deadtime value
```

**Syntax Description**

- `value` Length of time, in minutes, for which a RADIUS server is skipped over by transaction requests, up to a maximum of 1440 (24 hours). The range is from 1 to 1440. The default value is 0.

**Command Default**

Dead time is set to 0.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

A RADIUS server marked as dead is skipped by additional requests for the duration of minutes unless all other servers are marked dead and there is no rollover method.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

This example specifies five minutes of deadtime for RADIUS servers that fail to respond to authentication requests for the `radius-server deadtime` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# radius-server deadtime 5
```
radius-server key (BNG)

To set the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon, use the `radius-server key` command in the XR Config mode. To disable the key, use the `no` form of this command.

```
radius-server key {0 clear-text-key | 7 encrypted-key clear-text-key}
noradius-server key
```

**Syntax Description**
- **0 clear-text-key** Specifies an unencrypted (cleartext) shared key.
- **7 encrypted-key** Specifies an encrypted shared key.
- **clear-text-key** Specifies an unencrypted (cleartext) shared key.

**Command Default**
The authentication and encryption key is disabled.

**Command Modes**
XR Config mode

**Command History**
- **Release**
  - Release 6.0
  - Modification: This command was introduced.

**Usage Guidelines**
The key entered must match the key used on the RADIUS server. All leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

**Examples**
This example shows how to set the cleartext key to “samplekey”:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# radius-server key 0 samplekey
```

This example shows how to set the encrypted shared key to “anykey”:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# radius-server key 7 anykey
```
radius-server retransmit (BNG)

To specify the number of times the Cisco IOS XR software retransmits a packet to a server before giving up, use the `radius-server retransmit` command in the XR Config mode. The `no` form of this command sets it to the default value of 3.

```
radius-server retransmit { retries disable }
no radius-server retransmit { retries disable }
```

**Syntax Description**
- `retries` Maximum number of retransmission attempts. The range is from 1 to 100. Default is 3.
- `disable` Disables the radius-server transmit command.

**Command Default**
The RADIUS servers are retried three times, or until a response is received.

**Command Modes**
XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
The RADIUS client tries all servers, allowing each one to time out before increasing the retransmit count.

**Task ID**
```
Task ID  Operations
aaa      read, write
```

**Examples**
This example shows how to specify a retransmit counter value of five times:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# radius-server retransmit 5
```
radius-server timeout (BNG)

To set the interval for which a router waits for a server host to reply before timing out, use the `radius-server timeout` command in the XR Config mode. To restore the default, use the `no` form of this command.

```
radius-server timeout seconds
no radius-server timeout
```

**Syntax Description**

- `seconds` Number that specifies the timeout interval, in seconds. Range is from 1 to 1000.

**Command Default**

The default radius-server timeout value is 5 seconds.

**Command Modes**

XR Config mode

**Command History**

- **Release**
  - Release 6.0: This command was introduced.

**Usage Guidelines**

Use the `radius-server timeout` command to set the number of seconds a router waits for a server host to reply before timing out.

**Task ID**

- Task ID: aaa
  - Operations: read, write

**Examples**

This example shows how to change the interval timer to 10 seconds:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# radius-server timeout 10
```
radius source-interface (BNG)

To force RADIUS to use the IP address of a specified interface or subinterface for all outgoing RADIUS packets, use the `radius source-interface` command in the XR Config mode. To prevent only the specified interface from being the default and not from being used for all outgoing RADIUS packets, use the `no` form of this command.

```
radius source-interface interface [vrf vrf_name]
no radius source-interface interface
```

**Syntax Description**

- `interface-name` Name of the interface that RADIUS uses for all of its outgoing packets.
- `vrf vrf-name` Specifies the name of the assigned VRF.

**Command Default**

If a specific source interface is not configured, or the interface is down or does not have an IP address configured, the system selects an IP address.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `radius source-interface` command to set the IP address of the specified interface or subinterface for all outgoing RADIUS packets. This address is used as long as the interface or subinterface is in the up state. In this way, the RADIUS server can use one IP address entry for every network access client instead of maintaining a list of IP addresses.

The specified interface or subinterface must have an IP address associated with it. If the specified interface or subinterface does not have an IP address or is in the down state, then RADIUS reverts to the default. To avoid this, add an IP address to the interface or subinterface or bring the interface to the up state.

The `radius source-interface` command is especially useful in cases in which the router has many interfaces or subinterfaces and you want to ensure that all RADIUS packets from a particular router have the same IP address.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

This example shows how to make RADIUS use the IP address of subinterface s2 for all outgoing RADIUS packets:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# radius source-interface loopback 10 vrf vrf1
```
To configure an encrypted or clear-text password for the user, use the `secret` command in username configuration mode or line template configuration mode. To remove this configuration, use the `no` form of this command.

```
secret [{0 [enc-type enc-type-value] | 5 | 8 | 9 | 10}] secret-login
no secret
```

**Syntax Description**

- **0** (Optional) Specifies that an unencrypted (clear-text) password follows. The password will be encrypted for storage in the configuration using an MD5 encryption algorithm. Otherwise, the password is not encrypted.

- **5** Specifies that an encrypted MD5 password (secret) follows.

- **8** (Optional) Specifies that SHA256-encrypted password follows.

- **9** (Optional) Specifies that scrypt-encrypted password follows.

- **10** (Optional) Specifies that SHA512-encrypted password follows.

```
secret-login
```

Text string in alphanumeric characters that is stored as the MD5-encrypted password entered by the user in association with the user’s login ID.

- Can be up to 253 characters in length.

**Note** The characters entered must conform to MD5 encryption standards.

- **enc-type** (Optional) Configures the encryption type for a password entered in clear text.

- **enc-type-value** Specifies the encryption type to be used.

**Command Default**

No password is specified.

**Command Modes**

- Username configuration
- Line template configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>Release 7.0.1</td>
<td>Added the support for Type 8 (SHA256), Type 9 (scrypt) and Type 10 (SHA512) encryption for <code>secret</code> configuration.</td>
</tr>
<tr>
<td>Release 7.0.1</td>
<td>Added the support for <code>enc-type</code> option under <code>secret 0</code> to specify the type of encryption for password entered in clear-text format.</td>
</tr>
</tbody>
</table>
Usage Guidelines

From Release 7.0.1 and later, Type 10 encryption is applied as the default encryption type for the `secret` on Cisco IOS XR 64-bit operating systems. Prior to this, Type 5 (MD5) was the default one.

Prior to Release 7.0.1, Cisco IOS XR software allows you to configure only Message Digest 5 (MD5) encryption for username logins and passwords. MD5 encryption is a one-way hash function that makes reversal of an encrypted password impossible, providing strong encryption protection. Using MD5 encryption, you cannot retrieve clear-text passwords. Therefore, MD5 encrypted passwords cannot be used with protocols that require the clear-text password to be retrievable, such as Challenge Handshake Authentication Protocol (CHAP).

Prior to Release 7.0.1, you can specify only one of two types of secure secret IDs: encrypted (5) or clear text (0). If you do not select either 0 or 5, the clear-text password you enter is not encrypted.

When an XR EXEC mode process is started on a line that has password protection, the process prompts for the secret. If the user enters the correct secret, the process issues the prompt. The user can try entering the secret thrice before the terminal returns to the idle state.

Secrets are one-way encrypted and should be used for login activities that do not require a decryptable secret.

To verify that MD5 password encryption has been enabled, use the `show running-config` command. The “username name secret 5” line in the command output indicates the same.

The `show running-config` command does not display the login password in clear text when the **0** option is used to specify an unencrypted password. See the “Examples” section.

Note

The show running-config command does not display the login password in clear text when the **0** option is used to specify an unencrypted password. See the “Examples” section.

Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

Examples

The following example shows how to establish the clear-text secret “lab” for the user `user2`:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# username user2
RP/0/RP0/CPU0:router(config-un)# secret 0 lab
RP/0/RP0/CPU0:router(config-un)# commit
RP/0/RP0/CPU0:router(config-un)# show running-config
Building configuration...
username user2
secret 5 $1$DTmd$q7C6fhzje7Cc7Xzmu2Frx1
!
end
```

The following examples show how to configure a Type 10 (SHA512) password for the user, `user10`. You can also see the examples and usage of the `username, on page 95` command.

You can specify Type as ‘10’ under the `secret` keyword, to explicitly configure Type 10 password.

```
Router# configure
Router(config)# username user10 secret 10
$6$s90v03dlwveTEnqkAU9US3CL1Ej/F.E4v/tH.UaqTa0u9U3E9X98pG6c5p2hM3nEtg864j3k2Q7meAwyhu5VAv/aBF3jgj/jx2Gl7h6aPrwJWf1
Router(config-un)# commit
```
You can also use the `enc-type` keyword under the `secret 0` option, to specify Type 10 as the encryption for a password entered in clear text.

Router#configure
Router(config)#username user10 secret 0 enc-type 10 testpassword
Router(config-un)#commit
server (RADIUS)

To associate a particular RADIUS server with a defined server group, use the `server` command in RADIUS server-group configuration mode. To remove the associated server from the server group, use the `no` form of this command.

```
server  ip-address [auth-port port-number] [acct-port port-number]
no server  ip-address [auth-port port-number] [acct-port port-number]
```

**Syntax Description**

- `ip-address`: IP address of the RADIUS server host.
- `auth-port port-number`: (Optional) Specifies the User Datagram Protocol (UDP) destination port for authentication requests. The `port-number` argument specifies the port number for authentication requests. The host is not used for authentication if this value is set to 0. Default is 1645.
- `acct-port port-number`: (Optional) Specifies the UDP destination port for accounting requests. The `port-number` argument specifies the port number for accounting requests. The host is not used for accounting services if this value is set to 0. Default is 1646.

**Command Default**

If no port attributes are defined, the defaults are as follows:

- Authentication port: 1645
- Accounting port: 1646

**Command Modes**

RADIUS server-group configuration

**Command History**

- **Release**
  - Release 6.0: This command was introduced.

**Usage Guidelines**

Use the `server` command to associate a particular RADIUS server with a defined server group.

There are two different ways in which you can identify a server, depending on the way you want to offer AAA services. You can identify the server simply by using its IP address, or you can identify multiple host instances or entries using the optional `auth-port` and `acct-port` keywords.

When you use the optional keywords, the network access server identifies RADIUS security servers and host instances associated with a group server based on their IP address and specific UDP port numbers. The combination of the IP address and UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS host entries providing a specific AAA service. If two different host entries on the same RADIUS server are configured for the same service, for example, accounting, the second host entry configured acts as an automatic switchover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the network access server tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order they are configured.)
The following example shows how to use two different host entries on the same RADIUS server that are configured for the same services—authentication and accounting. The second host entry configured acts as switchover backup to the first one.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa group server radius group1
RP/0/RP0/CPU0:router(config-sg-radius)# server 1.1.1.1 auth-port 1645 acct-port 1646
RP/0/RP0/CPU0:router(config-sg-radius)# server 2.2.2.2 auth-port 2000 acct-port 2001
```
server (TACACS+)

To associate a particular TACACS+ server with a defined server group, use the `server` command in TACACS+ server-group configuration mode. To remove the associated server from the server group, use the `no` form of this command.

```
server    {hostname|ip-address}
no server {hostname|ip-address}
```

**Syntax Description**

- **hostname** Character string used to name the server host.
- **ip-address** IP address of the server host.

**Command Default**

None

**Command Modes**

TACACS+ server-group configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `server` command to associate a particular TACACS+ server with a defined server group. The server need not be accessible during configuration. Later, you can reference the configured server group from the method lists used to configure authentication, authorization, and accounting (AAA).

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to associate the TACACS+ server with the IP address 192.168.60.15 with the server group tac1:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa group server tacacs+ tac1
RP/0/RP0/CPU0:router(config-sg-tacacs+)# server 192.168.60.15
```
server-private (RADIUS)

To configure the IP address of the private RADIUS server for the group server, use the `server-private` command in RADIUS server-group configuration mode. To remove the associated private server from the AAA group server, use the `no` form of this command.

```
server-private ip-address [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]
no server-private ip-address [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]
```

### Syntax Description

- **ip-address**: IP address of the RADIUS server host.

- **auth-port port-number**: (Optional) Specifies the User Datagram Protocol (UDP) destination port for authentication requests. The `port-number` argument specifies the port number for authentication requests. The host is not used for authentication if this value is set to 0. The default value is 1645.

- **acct-port port-number**: (Optional) Specifies the UDP destination port for accounting requests. The `port-number` argument specifies the port number for accounting requests. The host is not used for accounting services if this value is set to 0. The default value is 1646.

- **timeout seconds**: (Optional) Specifies the number of seconds the router waits for the RADIUS server to reply before retransmitting. The setting overrides the global value of the `radius-server timeout` command. If no timeout is specified, the global value is used. The `seconds` argument specifies the timeout value in seconds. The range is from 1 to 1000. If no timeout is specified, the global value is used.

- **retransmit retries**: (Optional) Specifies the number of times a RADIUS request is resent to a server if the server is not responding or is responding slowly. The setting overrides the global setting of the `radius-server transmit` command. The `retries` argument specifies the retransmit value. The range is from 1 to 100. If no retransmit value is specified, the global value is used.

- **key string**: (Optional) Specifies the authentication and encryption key that is used between the router and the RADIUS daemon running on the RADIUS server. This key overrides the global setting of the `radius-server key` command. If no key string is specified, the global value is used.

### Command Default

If no port attributes are defined, the defaults are as follows:

- Authentication port: 1645
- Accounting port: 1646

### Command Modes

RADIUS server-group configuration

### Command History

**Release**

- Release 6.0

**Modification**

- This command was introduced.
Use the `server-private` command to associate a particular private server with a defined server group. Possible overlapping of IP addresses between VRF instances are permitted. Private servers (servers with private addresses) can be defined within the server group and remain hidden from other groups, while the servers in the global pool (for example, default radius server group) can still be referred to by IP addresses and port numbers. Thus, the list of servers in server groups includes references to the hosts in the configuration and the definitions of private servers.

Both the `auth-port` and `acct-port` keywords enter RADIUS server-group private configuration mode.

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read,</td>
<td>write</td>
</tr>
</tbody>
</table>

The following examples show how to define the group1 RADIUS group server, to associate private servers with it, and to enter RADIUS server-group private configuration mode:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa group server radius group1
RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 timeout 5
RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 retransmit 3
RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 key coke
RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 auth-port 300
RP/0/RP0/CPU0:router(config-sg-radius-private)# exit
RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.2.2.2 timeout 5
RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.2.2.2 retransmit 3
RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.2.2.2 key coke
RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.2.2.2 auth-port 300
RP/0/RP0/CPU0:router(config-sg-radius-private)#
```

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa group server radius group1
RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 auth-port 300
RP/0/RP0/CPU0:router(config-sg-radius-private)# exit
RP/0/RP0/CPU0:router(config-sg-radius)# server-private 10.2.2.2 auth-port 300
RP/0/RP0/CPU0:router(config-sg-radius-private)#
```
server-private (TACACS+)

To configure the IP address of the private TACACS+ server for the group server, use the `server-private` command in TACACS+ server-group configuration mode. To remove the associated private server from the AAA group server, use the `no` form of this command.

```
server-private {hostnameip-address} [port port-number] [timeout seconds] [key string]
no server-private {hostnameip-address}
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname</td>
<td>Character string used to name the server host.</td>
</tr>
<tr>
<td>ip-address</td>
<td>IP address of the TACACS+ server host. Both IPv4 and IPv6 addresses are supported.</td>
</tr>
<tr>
<td>port</td>
<td>(Optional) Specifies a server port number. This option overrides the default, which is port 49. Valid port numbers range from 1 to 65535.</td>
</tr>
<tr>
<td>port-number</td>
<td>(Optional) Specifies, in seconds, a timeout value that sets the length of time the authentication, authorization, and accounting (AAA) server waits to receive a response from the TACACS+ server. This option overrides the global timeout value set with the <code>tacacs-server timeout</code> command for only this server. The range is from 1 to 1000. The default is 5.</td>
</tr>
<tr>
<td>timeout</td>
<td>(Optional) Specifies the authentication and encryption key that is used between the router and the TACACS+ daemon running on the TACACS+ server. This key overrides the global setting of the <code>tacacs-server key</code> command. If no key string is specified, the global value is used.</td>
</tr>
</tbody>
</table>

**Command Default**

The `port-name` argument, if not specified, defaults to the standard port 49.

The `seconds` argument, if not specified, defaults to 5 seconds.

**Command Modes**

TACACS+ server-group configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `server-private` command to associate a particular private server with a defined server group. Possible overlapping of IP addresses between VRF instances are permitted. Private servers (servers with private addresses) can be defined within the server group and remain hidden from other groups, while the servers in the global pool (for example, default tacacs+ server group) can still be referred by IP addresses and port numbers. Therefore, the list of servers in server groups includes references to the hosts in the global configuration and the definitions of private servers.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>
This example shows how to define the myserver TACACS+ group server, to associate private servers with it, and to enter TACACS+ server-group private configuration mode:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa group server tacacs+ myserver
RP/0/RP0/CPU0:router(config-sg-tacacs+)# server-private 10.1.1.1 timeout 5
RP/0/RP0/CPU0:router(config-sg-tacacs+)# server-private 10.1.1.1 key a_secret
RP/0/RP0/CPU0:router(config-sg-tacacs+)# server-private 10.1.1.1 port 51
RP/0/RP0/CPU0:router(config-sg-tacacs-private)# exit
RP/0/RP0/CPU0:router(config-sg-tacacs+)# server-private 10.2.2.2 timeout 5
RP/0/RP0/CPU0:router(config-sg-tacacs+)# server-private 10.2.2.2 key coke
RP/0/RP0/CPU0:router(config-sg-tacacs+)# server-private 10.2.2.2 port 300
RP/0/RP0/CPU0:router(config-sg-tacacs-private)#
```
show aaa (XR-VM)

To display information about an Internet Key Exchange (IKE) Security Protocol group, user group, local user, login traces, or task group; to list all task IDs associated with all IKE groups, user groups, local users, or task groups in the system; or to list all task IDs for a specified IKE group, user group, local user, or task group, use the `show aaa` command in the XR EXEC mode.

```
show aaa {ikegroup ikegroup-name | login trace | usergroup [usergroup-name] | trace | userdb [username] | task supported | taskgroup [{root-lr | netadmin | operator | sysadmin | root-system | service-admin | cisco-support | taskgroup-name}}
```

### Syntax Description

- **ikegroup**: Displays details for all IKE groups.
  - **ikegroup-name**: (Optional) IKE group whose details are to be displayed.
- **login trace**: Displays trace data for login subsystem.
- **usergroup**: Displays details for all user groups.
- **root-lr**: (Optional) Usergroup name.
- **netadmin**: (Optional) Usergroup name.
- **operator**: (Optional) Usergroup name.
- **sysadmin**: (Optional) Usergroup name.
- **root-system**: (Optional) Usergroup name.
- **cisco-support**: (Optional) Usergroup name.
  - **usergroup-name**: (Optional) Usergroup name.
- **trace**: Displays trace data for AAA subsystem.
- **userdb**: Displays details for all local users and the usergroups to which each user belongs.
  - **username**: (Optional) User whose details are to be displayed.
- **task supported**: Displays all AAA task IDs available.
- **taskgroup**: Displays details for all task groups.
  - **Note**: For taskgroup keywords, see optional usergroup name keyword list.
  - **taskgroup-name**: (Optional) Task group whose details are to be displayed.

### Command Default

Details for all user groups, or all local users, or all task groups are listed if no argument is entered.

### Command Modes

XR EXEC mode
Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines

Use the `show aaa` command to list details for all IKE groups, user groups, local users, AAA task IDs, or task groups in the system. Use the optional `ikegroup-name`, `usergroup-name`, `username`, or `taskgroup-name` argument to display the details for a specified IKE group, user group, user, or task group, respectively.

Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read</td>
</tr>
</tbody>
</table>

Examples

The following sample output is from the `show aaa` command, using the `ikegroup` keyword:

```
RP/0/RP0/CPU0:router# show aaa ikegroup
IKE Group ike-group
  Max-Users = 50
IKE Group ikeuser
  Group-Rey = test-password
  Default Domain = cisco.com
IKE Group ike-user
```

The following sample output is from the `show aaa` command, using the `usergroup` command:

```
RP/0/RP0/CPU0:router# show aaa usergroup operator
User group 'operator'
  Inherits from task group 'operator'
User group 'operator' has the following combined set of task IDs (including all inherited groups):
  Task: basic-services : READ WRITE EXECUTE DEBUG
  Task: cdp : READ
  Task: diag : READ
  Task: ext-access : READ EXECUTE
  Task: logging : READ
```

The following sample output is from the `show aaa` command, using the `taskgroup` keyword for a task group named netadmin:

```
RP/0/RP0/CPU0:router# show aaa taskgroup netadmin
Task group 'netadmin'

Task group 'netadmin' has the following combined set of task IDs (including all inherited groups):
  Task: aaa : READ
  Task: acl : READ WRITE EXECUTE DEBUG
  Task: admin : READ
  Task: ancp : READ WRITE EXECUTE DEBUG
  Task: atm : READ WRITE EXECUTE DEBUG
  Task: basic-services : READ WRITE EXECUTE DEBUG
  Task: bcdl : READ
  Task: bfd : READ WRITE EXECUTE DEBUG
  Task: bgp : READ WRITE EXECUTE DEBUG
```
| Task:     | boot : READ WRITE EXECUTE DEBUG |
| Task:     | bundle : READ WRITE EXECUTE DEBUG |
| Task:     | cdp : READ WRITE EXECUTE DEBUG   |
| Task:     | cef : READ WRITE EXECUTE DEBUG   |
| Task:     | cgpn : READ WRITE EXECUTE DEBUG  |
| Task:     | config-mgmt : READ WRITE EXECUTE DEBUG |
| Task:     | config-services : READ WRITE EXECUTE DEBUG |
| Task:     | crypto : READ WRITE EXECUTE DEBUG |
| Task:     | diag : READ WRITE EXECUTE DEBUG  |
| Task:     | drivers : READ WRITE EXECUTE DEBUG |
| Task:     | dwdm : READ WRITE EXECUTE DEBUG  |
| Task:     | eem : READ WRITE EXECUTE DEBUG   |
| Task:     | ethernet-services : READ WRITE EXECUTE DEBUG |
| Task:     | ext-access : READ WRITE EXECUTE DEBUG |
| Task:     | fabric : READ WRITE EXECUTE DEBUG |
| Task:     | fault-mgr : READ WRITE EXECUTE DEBUG |
| Task:     | filesystem : READ WRITE EXECUTE DEBUG |
| Task:     | firewall : READ WRITE EXECUTE DEBUG |
| Task:     | fr : READ WRITE EXECUTE DEBUG    |
| Task:     | hdlc : READ WRITE EXECUTE DEBUG  |
| Task:     | host-services : READ WRITE EXECUTE DEBUG |
| Task:     | hsrp : READ WRITE EXECUTE DEBUG  |
| Task:     | interface : READ WRITE EXECUTE DEBUG |
| Task:     | inventory : READ WRITE EXECUTE DEBUG |
| Task:     | ip-services : READ WRITE EXECUTE DEBUG |
| Task:     | ipv4 : READ WRITE EXECUTE DEBUG  |
| Task:     | ipv6 : READ WRITE EXECUTE DEBUG  |
| Task:     | isis : READ WRITE EXECUTE DEBUG  |
| Task:     | l2vpn : READ WRITE EXECUTE DEBUG |
| Task:     | li : READ WRITE EXECUTE DEBUG    |
| Task:     | logging : READ WRITE EXECUTE DEBUG |
| Task:     | lpts : READ WRITE EXECUTE DEBUG  |
| Task:     | monitor : READ WRITE EXECUTE DEBUG |
| Task:     | mpls-ldp : READ WRITE EXECUTE DEBUG |
| Task:     | mpls-static : READ WRITE EXECUTE DEBUG |
| Task:     | mpls-te : READ WRITE EXECUTE DEBUG |
| Task:     | multicast : READ WRITE EXECUTE DEBUG |
| Task:     | netflow : READ WRITE EXECUTE DEBUG |
| Task:     | network : READ WRITE EXECUTE DEBUG |
| Task:     | ospf : READ WRITE EXECUTE DEBUG  |
| Task:     | ouni : READ WRITE EXECUTE DEBUG  |
| Task:     | pkg-mgmt : READ WRITE EXECUTE DEBUG |
| Task:     | ppp : READ WRITE EXECUTE DEBUG   |
| Task:     | qos : READ WRITE EXECUTE DEBUG   |
| Task:     | rib : READ WRITE EXECUTE DEBUG   |
| Task:     | rip : READ WRITE EXECUTE DEBUG   |
| Task:     | root-lr : READ WRITE EXECUTE DEBUG (reserved) |
| Task:     | route-map : READ WRITE EXECUTE DEBUG |
| Task:     | route-policy : READ WRITE EXECUTE DEBUG |
| Task:     | sbc : READ WRITE EXECUTE DEBUG   |
| Task:     | snmp : READ WRITE EXECUTE DEBUG  |
| Task:     | sonet-sdh : READ WRITE EXECUTE DEBUG |
| Task:     | static : READ WRITE EXECUTE DEBUG |
| Task:     | sysmgr : READ WRITE EXECUTE DEBUG |
| Task:     | system : READ WRITE EXECUTE DEBUG |
| Task:     | transport : READ WRITE EXECUTE DEBUG |
| Task:     | tty-access : READ WRITE EXECUTE DEBUG |
| Task:     | tunnel : READ WRITE EXECUTE DEBUG |
| Task:     | universal : READ WRITE EXECUTE DEBUG (reserved) |
| Task:     | vlan : READ WRITE EXECUTE DEBUG  |
| Task:     | vrrp : READ WRITE EXECUTE DEBUG   |
The following sample output is from the `show aaa` command, using the `taskgroup` keyword for an operator. The task group operator has the following combined set of task IDs, which includes all inherited groups:

<table>
<thead>
<tr>
<th>Task</th>
<th>IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic-services</td>
<td>READ WRITE EXECUTE DEBUG</td>
</tr>
<tr>
<td>cdp</td>
<td>READ</td>
</tr>
<tr>
<td>diag</td>
<td>READ</td>
</tr>
<tr>
<td>ext-access</td>
<td>READ EXECUTE</td>
</tr>
<tr>
<td>logging</td>
<td>READ</td>
</tr>
</tbody>
</table>

The following sample output is from the `show aaa` task group displaying the different task groups:

<table>
<thead>
<tr>
<th>Task</th>
<th>IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>READ</td>
</tr>
<tr>
<td>acl</td>
<td>READ</td>
</tr>
<tr>
<td>admin</td>
<td>READ</td>
</tr>
<tr>
<td>basic-services</td>
<td>READ</td>
</tr>
<tr>
<td>boot</td>
<td>READ</td>
</tr>
<tr>
<td>cisco-support</td>
<td>READ (reserved)</td>
</tr>
<tr>
<td>config-mgmt</td>
<td>READ</td>
</tr>
<tr>
<td>config-services</td>
<td>READ</td>
</tr>
<tr>
<td>crypto</td>
<td>READ</td>
</tr>
<tr>
<td>dwdm</td>
<td>READ</td>
</tr>
<tr>
<td>ethernet-services</td>
<td>READ</td>
</tr>
<tr>
<td>fabric</td>
<td>READ</td>
</tr>
<tr>
<td>fault-mgr</td>
<td>READ</td>
</tr>
<tr>
<td>filesystem</td>
<td>READ</td>
</tr>
<tr>
<td>hdlc</td>
<td>READ</td>
</tr>
<tr>
<td>host-services</td>
<td>READ</td>
</tr>
<tr>
<td>hsrp</td>
<td>READ</td>
</tr>
<tr>
<td>interface</td>
<td>READ</td>
</tr>
<tr>
<td>inventory</td>
<td>READ</td>
</tr>
<tr>
<td>ip-services</td>
<td>READ</td>
</tr>
<tr>
<td>ipv4</td>
<td>READ</td>
</tr>
<tr>
<td>ipv6</td>
<td>READ</td>
</tr>
<tr>
<td>logging</td>
<td>READ</td>
</tr>
<tr>
<td>mpls-te</td>
<td>READ</td>
</tr>
</tbody>
</table>

The following sample output is from `show aaa` command with the `userdb` keyword:

```
RP/0/RP0/CPU0:router# show aaa userdb
Username lab (admin plane)
User group root-system
User group cisco-support
Username acme
User group root-system
```

The following sample output is from the `show aaa` command, using the `task supported` keywords. Task IDs are displayed in alphabetic order.

```
RP/0/RP0/CPU0:router# show aaa task supported
aaa
acl
admin
atm
basic-services
bcdl
bfd
bgp
boot
```
bundle
cdp
ces
Cisco-Support
config-mgmt
config-services
crypto
diag
disallowed
drivers
ext-access
fabric
fault-mgr
filesystem
firewall
fr
hdlc
host-services
hsrp
interface
inventory
ip-services
ipv4
ipv6
isis
logging
lpts
monitor
mpls-ldp
mpls-static
mpls-te
multicast
netflow
network
ospf
ouni
pkg-mgmt

ppp
go
rib
rip
User group root-system
root-system
route-map
route-policy
sbc
snmp
sonet-sdh
static
sysmgr
system
transport
tty-access
tunnel
universal
vlan
vrrp
show aaa accounting

To display command history with the date and time for AAA sub-system, use the show aaa accounting command in the System Admin EXEC mode. You must have a group aaa-r or root-system on System Admin VM.

show aaa accounting

Syntax Description
This command has no keywords or arguments.

Command Default
None

Command Modes
System Admin EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines
No specific guidelines impact the use of this command.

Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read</td>
</tr>
</tbody>
</table>

This is the sample output of the show aaa accounting command:

```
sysadmin-vm:0_RP0#show aaa accounting
Mon Nov 3 13:37:21.573 UTC
Detail audit log information
Time Username Session-ID Node-Information Command
---------------------------------------------------------------------------------------------------
2014-11-03 13:14:27 UTC root 17 System logged in from the CLI with aaa disabled
.. 2014-11-03 13:37:01 UTC cisco 57 0/RP0 assigned to groups: root-system
2014-11-03 13:37:03 UTC cisco 57 0/RP0 terminal'
2014-11-03 13:37:03 UTC cisco 57 0/RP0 CLI 'config
2014-11-03 13:37:09 UTC cisco 57 0/RP0 CLI done
2014-11-03 13:37:11 UTC cisco 57 0/RP0 authentication users user temp'
2014-11-03 13:37:09 UTC cisco 57 0/RP0 CLI done
2014-11-03 13:37:11 UTC cisco 57 0/RP0 CLI 'password
*** 2014-11-03 13:37:11 UTC cisco 57 0/RP0 CLI done
2014-11-03 13:37:12 UTC cisco 57 0/RP0 CLI 'commit'
2014-11-03 13:37:14 UTC cisco 57 0/RP0 CLI done
2014-11-03 13:37:16 UTC cisco 57 0/RP0 CLI 'exit'
2014-11-03 13:37:16 UTC cisco 57 0/RP0 CLI done
2014-11-03 13:37:18 UTC cisco 57 0/RP0 CLI 'exit'
2014-11-03 13:37:18 UTC cisco 57 0/RP0 CLI done
```
show aaa accounting

2014-11-03 13:37:21 UTC  cisco  57  0/RP0  CLI 'show aaa accounting'
show aaa password-policy

To display the details of AAA password policy configured in a system, use the `show aaa password-policy` command in XR EXEC mode.

```
show aaa password-policy [policy-name]
```

**Syntax Description**

- **policy-name**: Specifies the name of password policy.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>6.2.1</td>
<td></td>
</tr>
</tbody>
</table>

**Usage Guidelines**

If the option `policy-name` is not specified, the command output displays the details of all password policies configured in the system.

Refer `aaa password-policy` command details of each field in this command output.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read</td>
</tr>
</tbody>
</table>

This is a sample output of `show aaa password-policy` command:

```
RP/0/RP0/CPU0:router#show aaa password-policy test-policy
Fri Feb 3 16:50:58.086 EDT
Password Policy Name : test-policy
  Number of Users : 1
  Minimum Length : 2
  Maximum Length : 253
  Special Character Len : 0
  Uppercase Character Len : 0
  Lowercase Character Len : 1
  Numeric Character Len : 0
  Policy Life Time :
    seconds : 0
    minutes : 0
    hours : 0
    days : 0
    months : 0
    years : 0
  Lockout Time :
    seconds : 0
    minutes : 0
    hours : 0
    days : 0
```
show aaa password-policy

- months: 0
- years: 0
- Character Change Len: 4
- Maximum Failure Attempts: 0

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa password-policy, on page 18</td>
<td>Defines the FIPS-compliant AAA password security policy.</td>
</tr>
</tbody>
</table>
show radius

To display information about the RADIUS servers that are configured in the system, use the `show radius` command in the XR EXEC mode.

**show radius**

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

If no radius servers are configured, no output is displayed.

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `show radius` command to display statistics for each configured RADIUS server.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaaa</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**

The following sample output is for the `show radius` command:

```
RP/0/RP0/CPU0:router# show radius

Global dead time: 0 minute(s)

Server: 1.1.1.1/1645/1646 is UP
  Timeout: 5 sec, Retransmit limit: 3
  Quarantined: No
  Authentication:
    0 requests, 0 pending, 0 retransmits
    0 accepts, 0 rejects, 0 challenges
    0 timeouts, 0 bad responses, 0 bad authenticators
    0 unknown types, 0 dropped, 0 ms latest rtt
  Accounting:
    0 requests, 0 pending, 0 retransmits
    0 responses, 0 timeouts, 0 bad responses
    0 bad authenticators, 0 unknown types, 0 dropped
    0 ms latest rtt

Server: 2.2.2.2/1645/1646 is UP
  Timeout: 10 sec, Retransmit limit: 3
  Authentication:
    0 requests, 0 pending, 0 retransmits
    0 accepts, 0 rejects, 0 challenges
    0 timeouts, 0 bad responses, 0 bad authenticators
    0 unknown types, 0 dropped, 0 ms latest rtt
  Accounting:
    0 requests, 0 pending, 0 retransmits
```
This table describes the significant fields shown in the display.

### Table 2: `show radius` Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Server IP address/UDP destination port for authentication requests/UDP destination port for accounting requests.</td>
</tr>
<tr>
<td>Timeout</td>
<td>Number of seconds the router waits for a server host to reply before timing out.</td>
</tr>
<tr>
<td>Retransmit limit</td>
<td>Number of times the Cisco IOS XR software searches the list of RADIUS server hosts before giving up.</td>
</tr>
</tbody>
</table>
show radius accounting

To obtain information and detailed statistics for the RADIUS accounting server and port, use the show radius accounting command in the XR EXEC mode.

Syntax Description
This command has no keywords or arguments.

Command Default
If no RADIUS servers are configured on the router, the output is empty. If the default values are for the counter (for example, request and pending), the values are all zero because the RADIUS server was just defined and not used yet.

Command Modes
XR EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines
No specific guidelines impact the use of this command.

Task ID
<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read</td>
</tr>
</tbody>
</table>

Examples
The following sample output is displayed on a per-server basis for the show radius accounting command:

```
RP/0/RP0/CPU0:router# show radius accounting
Server: 12.26.25.61, port: 1813
  0 requests, 0 pending, 0 retransmits
  0 responses, 0 timeouts, 0 bad responses
  0 bad authenticators, 0 unknown types, 0 dropped
  0 ms latest rtt

Server: 12.26.49.12, port: 1813
  0 requests, 0 pending, 0 retransmits
  0 responses, 0 timeouts, 0 bad responses
  0 bad authenticators, 0 unknown types, 0 dropped
  0 ms latest rtt

Server: 12.38.28.18, port: 29199
  0 requests, 0 pending, 0 retransmits
  0 responses, 0 timeouts, 0 bad responses
  0 bad authenticators, 0 unknown types, 0 dropped
  0 ms latest rtt
```

This table describes the significant fields shown in the display.
Table 3: show radius accounting Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Server IP address/UDP destination port for authentication requests; UDP destination port for accounting requests.</td>
</tr>
</tbody>
</table>
show radius authentication

To obtain information and detailed statistics for the RADIUS authentication server and port, use the show radius authentication command in the XR EXEC mode.

show radius authentication

Syntax Description
This command has no keywords or arguments.

Command Default
If no RADIUS servers are configured on the router, the output is empty. If the default values are for the counter (for example, request and pending), the values are all zero because the RADIUS server was just defined and not used yet.

Command Modes
XR EXEC mode

Command History
Release               Modification
Release 6.0            This command was introduced.

Usage Guidelines
No specific guidelines impact the use of this command.

Task ID
<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read</td>
</tr>
</tbody>
</table>

Examples
The following sample output is for the show radius authentication command:

RP/0/RP0/CPU0:router# show radius authentication

Server: 12.26.25.61, port: 1812
0 requests, 0 pending, 0 retransmits
0 accepts, 0 rejects, 0 challenges
0 timeouts, 0 bad responses, 0 bad authenticators
0 unknown types, 0 dropped, 0 ms latest rtt

Server: 12.26.49.12, port: 1812
0 requests, 0 pending, 0 retransmits
0 accepts, 0 rejects, 0 challenges
0 timeouts, 0 bad responses, 0 bad authenticators
0 unknown types, 0 dropped, 0 ms latest rtt

Server: 12.38.28.18, port: 21099
0 requests, 0 pending, 0 retransmits
0 accepts, 0 rejects, 0 challenges
0 timeouts, 0 bad responses, 0 bad authenticators
0 unknown types, 0 dropped, 0 ms latest rtt

This table describes the significant fields shown in the display.
Table 4: show radius authentication Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Server IP address/UDP destination port for authentication requests; UDP destination port for accounting requests.</td>
</tr>
</tbody>
</table>
show radius dead-criteria

To obtain information about the dead server detection criteria, use the `show radius dead-criteria` command in the XR EXEC mode.

```
show radius dead-criteria host ip-addr [auth-port auth-port] [acct-port acct-port]
```

**Syntax Description**

- `host ip-addr` (Optional) Specifies the name or IP address of the configured RADIUS server.
- `auth-port auth-port` (Optional) Specifies the authentication port for the RADIUS server. The default value is 1645.
- `acct-port acct-port` (Optional) Specifies the accounting port for the RADIUS server. The default value is 1646.

**Command Default**

The default values for time and tries are not fixed to a single value; therefore, they are calculated and fall within a range of 10 to 60 seconds for time and 10 to 100 for tries.

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

- aaa read

**Examples**

The following sample output is for the `show radius dead-criteria` command:

```
RP/0/RP0/CPU0:router# show radius dead-criteria host 12.26.49.12 auth-port 11000 acct-port 11001
Server: 12.26.49.12/11000/11001
Dead criteria time: 10 sec (computed) tries: 10 (computed)
```

This table describes the significant fields shown in the display.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Server IP address/UDP destination port for authentication requests/UDP destination port for accounting requests.</td>
</tr>
<tr>
<td>Timeout</td>
<td>Number of seconds the router waits for a server host to reply before timing out.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Retransmits</td>
<td>Number of times Cisco IOS XR software searches the list of RADIUS server hosts before giving up.</td>
</tr>
</tbody>
</table>
show radius server-groups

To display information about the RADIUS server groups that are configured in the system, use the `show radius server-groups` command in the XR EXEC mode.

```
show radius server-groups [group-name [detail]]
```

**Syntax Description**

- `group-name` (Optional) Name of the server group. The properties are displayed.
- `detail` (Optional) Displays properties for all the server groups.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `show radius server-groups` command to display information about each configured RADIUS server group, including the group name, numbers of servers in the group, and a list of servers in the named server group. A global list of all configured RADIUS servers, along with authentication and accounting port numbers, is also displayed.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**

The inherited global message is displayed if no group level deadtime is defined for this group; otherwise, the group level deadtime value is displayed and this message is omitted. The following sample output is for the `show radius server-groups` command:

```
RP/0/RP0/CPU0:router# show radius server-groups

Global list of servers
Contains 2 server(s)
  Server 1.1.1.1/1645/1646
  Server 2.2.2.2/1645/1646

Server group 'radgrp1' has 2 server(s)
  Dead time: 0 minute(s) (inherited from global)
  Contains 2 server(s)
    Server 1.1.1.1/1645/1646
    Server 2.2.2.2/1645/1646

Server group 'radgrp-priv' has 1 server(s)
  Dead time: 0 minute(s) (inherited from global)
  Contains 1 server(s)
    Server 3.3.3.3/1645/1646 [private]
```
The following sample output shows the properties for all the server groups in group “radgrp1:”

RP/0/RP0/CPU0:router# show radius server-groups radgrp1 detail

Server group 'radgrp1' has 2 server(s)
VRF default (id 0x60000000)
Dead time: 0 minute(s) (inherited from global)
Contains 2 server(s)
  Server 1.1.1.1/1645/1646
    Authentication:
      0 requests, 0 pending, 0 retransmits
      0 accepts, 0 rejects, 0 challenges
      0 timeouts, 0 bad responses, 0 bad authenticators
      0 unknown types, 0 dropped, 0 ms latest rtt
    Accounting:
      0 requests, 0 pending, 0 retransmits
      0 responses, 0 timeouts, 0 bad responses
      0 bad authenticators, 0 unknown types, 0 dropped
      0 ms latest rtt
  Server 2.2.2.2/1645/1646
    Authentication:
      0 requests, 0 pending, 0 retransmits
      0 accepts, 0 rejects, 0 challenges
      0 timeouts, 0 bad responses, 0 bad authenticators
      0 unknown types, 0 dropped, 0 ms latest rtt
    Accounting:
      0 requests, 0 pending, 0 retransmits
      0 responses, 0 timeouts, 0 bad responses
      0 bad authenticators, 0 unknown types, 0 dropped
      0 ms latest rtt

The following sample output shows the properties for all the server groups in detail in the group “raddgrp-priv:”

RP/0/RP0/CPU0:router# show radius server-groups radgrp-priv detail

Server group 'radgrp-priv' has 1 server(s)
VRF default (id 0x60000000)
Dead time: 0 minute(s) (inherited from global)
Contains 1 server(s)
  Server 3.3.3.3/1645/1646 [private]
    Authentication:
      0 requests, 0 pending, 0 retransmits
      0 accepts, 0 rejects, 0 challenges
      0 timeouts, 0 bad responses, 0 bad authenticators
      0 unknown types, 0 dropped, 0 ms latest rtt
    Accounting:
      0 requests, 0 pending, 0 retransmits
      0 responses, 0 timeouts, 0 bad responses
      0 bad authenticators, 0 unknown types, 0 dropped
      0 ms latest rtt

This table describes the significant fields shown in the display.

Table 6: show radius server-groups Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Server IP address/UDP destination port for authentication requests/UDP destination port for accounting requests.</td>
</tr>
</tbody>
</table>
show tacacs

To display information about the TACACS+ servers that are configured in the system, use the `show tacacs` command in the XR EXEC mode.

```plaintext
show tacacs
```

**Syntax Description**
This command has no keywords or arguments.

**Command Default**
None

**Command Modes**
XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
Use the `show tacacs` command to display statistics for each configured TACACS+ server.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**
The following is sample output from the `show tacacs` command:

```
RP/0/RP0/CPU0:router# show tacacs
For IPv4 IP addresses:
Server: 1.1.1.1/21212 opens=0 closes=0 aborts=0 errors=0
        packets in=0 packets out=0
        status=up single-connect=false
Server: 2.2.2.2/21232 opens=0 closes=0 aborts=0 errors=0
        packets in=0 packets out=0
        status=up single-connect=false

For IPv6 IP addresses:
Server: 1.2.3.4/49 family = AF_INET opens=0 closes=0 aborts=0 errors=0
        packets in=0 packets out=0
        status=up single-connect=false
```

This table describes the significant fields shown in the display.

**Table 7: show tacacs Field Descriptions**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Server IP address.</td>
</tr>
<tr>
<td>opens</td>
<td>Number of socket opens to the external server.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>closes</td>
<td>Number of socket closes to the external server.</td>
</tr>
<tr>
<td>aborts</td>
<td>Number of tacacs requests that have been terminated midway.</td>
</tr>
<tr>
<td>errors</td>
<td>Number of error replies from the external server.</td>
</tr>
<tr>
<td>packets in</td>
<td>Number of TCP packets that have been received from the external server.</td>
</tr>
<tr>
<td>packets out</td>
<td>Number of TCP packets that have been sent to the external server.</td>
</tr>
</tbody>
</table>
show tacacs server-groups

To display information about the TACACS+ server groups that are configured in the system, use the show tacacs server-groups command in the XR EXEC mode.

show tacacs server-groups

Syntax Description

This command has no keywords or arguments.

Command Default

None

Command Modes

XR EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines

Use the show tacacs server-groups command to display information about each configured TACACS+ server group, including the group name, numbers of servers in the group, and a list of servers in the named server group. A global list of all configured TACACS+ servers is also displayed.

Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read</td>
</tr>
</tbody>
</table>

Examples

The following is sample output from the show tacacs server-groups command:

```
RP/0/RP0/CPU0:router# show tacacs server-groups

Global list of servers
Server 12.26.25.61/23456
Server 12.26.49.12/12345
Server 12.26.49.12/9000
Server 12.26.25.61/23432
Server 5.5.5.5/23456
Server 1.1.1.1/49
Server group ‘tac100’ has 1 servers
Server 12.26.49.12
```

This table describes the significant fields shown in the display.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Server IP address.</td>
</tr>
</tbody>
</table>
show user

To display all user groups and task IDs associated with the currently logged-in user, use the `show user` command in the XR EXEC mode.

```
show user [{all | authentication | group | tasks}]
```

**Syntax Description**

- **all**: (Optional) Displays all user groups and task IDs for the currently logged-in user.
- **authentication**: (Optional) Displays authentication method parameters for the currently logged-in user.
- **group**: (Optional) Displays the user groups associated with the currently logged-in user.
- **tasks**: (Optional) Displays task IDs associated with the currently logged-in user. The **tasks** keyword indicates which task is reserved in the sample output.

**Command Default**

When the `show user` command is used without any option, it displays the ID of the user who is logged in currently.

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `show user` command to display all user groups and task IDs associated with the currently logged-in user.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>—</td>
</tr>
</tbody>
</table>

**Examples**

The following sample output displays the authentication method parameters from the `show user` command:

```
RP/0/RP0/CPU0:router# show user authentication method
local
```

The following sample output displays the groups from the `show user` command:

```
RP/0/RP0/CPU0:router# show user group
root-system
```

The following sample output displays all the information for the groups and tasks from the `show user` command:
show user all
Username: lab
Groups: root-system
Authenticated using method local
User lab has the following Task ID(s):

Task: aaa : READ WRITE EXECUTE DEBUG
Task: acl : READ WRITE EXECUTE DEBUG
Task: admin : READ WRITE EXECUTE DEBUG
Task: atm : READ WRITE EXECUTE DEBUG
Task: basic-services : READ WRITE EXECUTE DEBUG
Task: bcdl : READ WRITE EXECUTE DEBUG
Task: bfd : READ WRITE EXECUTE DEBUG
Task: bgp : READ WRITE EXECUTE DEBUG
Task: boot : READ WRITE EXECUTE DEBUG
Task: bundle : READ WRITE EXECUTE DEBUG
Task: cdp : READ WRITE EXECUTE DEBUG
Task: cef : READ WRITE EXECUTE DEBUG
Task: config-mgmt : READ WRITE EXECUTE DEBUG
Task: config-services : READ WRITE EXECUTE DEBUG
Task: crypto : READ WRITE EXECUTE DEBUG
Task: diag : READ WRITE EXECUTE DEBUG
Task: drivers : READ WRITE EXECUTE DEBUG
Task: ext-access : READ WRITE EXECUTE DEBUG
Task: fabric : READ WRITE EXECUTE DEBUG
Task: fault-mgr : READ WRITE EXECUTE DEBUG
Task: filesystem : READ WRITE EXECUTE DEBUG
Task: firewall : READ WRITE EXECUTE DEBUG
Task: fr : READ WRITE EXECUTE DEBUG
Task: hdic : READ WRITE EXECUTE DEBUG
Task: host-services : READ WRITE EXECUTE DEBUG
Task: hsrp : READ WRITE EXECUTE DEBUG
Task: interface : READ WRITE EXECUTE DEBUG
Task: inventory : READ WRITE EXECUTE DEBUG
Task: ip-services : READ WRITE EXECUTE DEBUG
Task: ipv4 : READ WRITE EXECUTE DEBUG
Task: ipv6 : READ WRITE EXECUTE DEBUG
Task: isis : READ WRITE EXECUTE DEBUG
Task: logging : READ WRITE EXECUTE DEBUG
Task: lpts : READ WRITE EXECUTE DEBUG
Task: monitor : READ WRITE EXECUTE DEBUG
Task: mpls-ldp : READ WRITE EXECUTE DEBUG
Task: mpls-static : READ WRITE EXECUTE DEBUG
Task: mpls-te : READ WRITE EXECUTE DEBUG
Task: multicast : READ WRITE EXECUTE DEBUG
Task: netflow : READ WRITE EXECUTE DEBUG
Task: network : READ WRITE EXECUTE DEBUG
Task: ospf : READ WRITE EXECUTE DEBUG
Task: ouni : READ WRITE EXECUTE DEBUG
Task: pkg-mgmt : READ WRITE EXECUTE DEBUG
Task: ppp : READ WRITE EXECUTE DEBUG
Task: qos : READ WRITE EXECUTE DEBUG
Task: rib : READ WRITE EXECUTE DEBUG
Task: rip : READ WRITE EXECUTE DEBUG
Task: root-1r : READ WRITE EXECUTE DEBUG (reserved)
Task: root-system : READ WRITE EXECUTE DEBUG (reserved)
Task: route-map : READ WRITE EXECUTE DEBUG
Task: route-policy : READ WRITE EXECUTE DEBUG
Task: sbc : READ WRITE EXECUTE DEBUG
Task: snmp : READ WRITE EXECUTE DEBUG
Task: sonet-sdh : READ WRITE EXECUTE DEBUG
Task: static : READ WRITE EXECUTE DEBUG
Task: sysmgr : READ WRITE EXECUTE DEBUG
Task: system : READ WRITE EXECUTE DEBUG
Task: transport : READ WRITE EXECUTE DEBUG
Task: tty-access : READ WRITE EXECUTE DEBUG
Task: tunnel : READ WRITE EXECUTE DEBUG
Task: universal : READ WRITE EXECUTE DEBUG (reserved)
Task: vlan : READ WRITE EXECUTE DEBUG
Task: vrrp : READ WRITE EXECUTE DEBUG

The following sample output displays the tasks and indicates which tasks are reserved from the `show user` command:

```
RP/0/RP0/CPU0:router# show user tasks
Task: aaa : READ WRITE EXECUTE DEBUG
Task: acl : READ WRITE EXECUTE DEBUG
Task: admin : READ WRITE EXECUTE DEBUG
Task: atm : READ WRITE EXECUTE DEBUG
Task: basic-services : READ WRITE EXECUTE DEBUG
Task: bfd : READ WRITE EXECUTE DEBUG
Task: bgp : READ WRITE EXECUTE DEBUG
Task: boot : READ WRITE EXECUTE DEBUG
Task: bundle : READ WRITE EXECUTE DEBUG
Task: cdp : READ WRITE EXECUTE DEBUG
Task: cef : READ WRITE EXECUTE DEBUG
Task: config-mgmt : READ WRITE EXECUTE DEBUG
Task: config-services : READ WRITE EXECUTE DEBUG
Task: crypto : READ WRITE EXECUTE DEBUG
Task: diag : READ WRITE EXECUTE DEBUG
Task: drivers : READ WRITE EXECUTE DEBUG
Task: ext-access : READ WRITE EXECUTE DEBUG
Task: fabric : READ WRITE EXECUTE DEBUG
Task: fault-mgr : READ WRITE EXECUTE DEBUG
Task: filesystem : READ WRITE EXECUTE DEBUG
Task: firewall : READ WRITE EXECUTE DEBUG
Task: fr : READ WRITE EXECUTE DEBUG
Task: hdlc : READ WRITE EXECUTE DEBUG
Task: host-services : READ WRITE EXECUTE DEBUG
Task: herp : READ WRITE EXECUTE DEBUG
Task: interface : READ WRITE EXECUTE DEBUG
Task: inventory : READ WRITE EXECUTE DEBUG
Task: ip-services : READ WRITE EXECUTE DEBUG
Task: ipv4 : READ WRITE EXECUTE DEBUG
Task: ipv6 : READ WRITE EXECUTE DEBUG
Task: isis : READ WRITE EXECUTE DEBUG
Task: logging : READ WRITE EXECUTE DEBUG
Task: lpts : READ WRITE EXECUTE DEBUG
Task: monitor : READ WRITE EXECUTE DEBUG
Task: mpls-ldp : READ WRITE EXECUTE DEBUG
Task: mpls-static : READ WRITE EXECUTE DEBUG
Task: mpls-te : READ WRITE EXECUTE DEBUG
Task: multicast : READ WRITE EXECUTE DEBUG
Task: netflow : READ WRITE EXECUTE DEBUG
Task: network : READ WRITE EXECUTE DEBUG
Task: ospf : READ WRITE EXECUTE DEBUG
Task: ouni : READ WRITE EXECUTE DEBUG
Task: pkg-mgmt : READ WRITE EXECUTE DEBUG
Task: ppp : READ WRITE EXECUTE DEBUG
Task: qos : READ WRITE EXECUTE DEBUG
Task: rib : READ WRITE EXECUTE DEBUG
Task: rip : READ WRITE EXECUTE DEBUG
```
<table>
<thead>
<tr>
<th>Task:</th>
<th>root-lr</th>
<th>READ</th>
<th>WRITE</th>
<th>EXECUTE</th>
<th>DEBUG (reserved)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task:</td>
<td>root-system</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG (reserved)</td>
</tr>
<tr>
<td>Task:</td>
<td>route-map</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>route-policy</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>sbc</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>snmp</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>sonet-sdh</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>static</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>syslog</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>system</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>transport</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>tty-access</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>tunnel</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>universal</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG (reserved)</td>
</tr>
<tr>
<td>Task:</td>
<td>vlan</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Task:</td>
<td>vrrp</td>
<td>READ</td>
<td>WRITE</td>
<td>EXECUTE</td>
<td>DEBUG</td>
</tr>
</tbody>
</table>
show aaa user-group

To display user group information for AAA sub-system, use the show aaa user-group command in the System Admin EXEC mode. You must have a group aaa-r or root-system on System Admin VM.

**show aaa user-group**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>This command has no keywords or arguments.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Default</td>
<td>None</td>
</tr>
<tr>
<td>Command Modes</td>
<td>System Admin EXEC mode</td>
</tr>
<tr>
<td>Command History</td>
<td>Release 6.0</td>
</tr>
<tr>
<td></td>
<td>Modification</td>
</tr>
<tr>
<td></td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>Usage Guidelines</td>
<td>No specific guidelines impact the use of this command.</td>
</tr>
</tbody>
</table>

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read</td>
</tr>
</tbody>
</table>

This is the sample output of the show aaa user-group command:

```
sysadmin-vm:0_RP0#show aaa user-group
Mon Nov 3 13:39:33.380 UTC
User group : root-system
sysadmin-vm:0_RP0#
```
show tech-support aaa

To collect AAA debug and trace files from System Admin VM, use the `show tech-support aaa` command in the System Admin EXEC mode.

```
show tech-support aaa
```

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

System Admin EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read</td>
</tr>
</tbody>
</table>

This is the sample output of the `show tech-support aaa` command:

```
sysadmin-vm:0_RP0#show tech-support aaa
Mon Nov 3 13:39:33.380 UTC
Waiting for gathering to complete /opt/cisco/calvados/script/show_tech_aaa: line 27: rse:
command not found.
Compressing show tech output
Show tech output available at /misc/disk1//showtech-aaa-admin-2014-Nov-04.082457.UTC.tgz
Please collect show tech-support ctrace in addition to any sysadmin show-tech-support collection
++ Show tech end time: 2014-Nov-04.UTC ++
sysadmin-vm:0_RP0#
```
single-connection

To multiplex all TACACS+ requests to this server over a single TCP connection, use the `single-connection` command in TACACS host configuration mode. To disable the single TCP connection for all new sessions that use a separate connection, use the `no` form of this command.

```
single-connection
no single-connection
```

**Syntax Description**
This command has no keywords or arguments.

**Command Default**
By default, a separate connection is used for each session.

**Command Modes**
TACACS host configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
The `single-connection` command allows the TACACS+ server to handle a greater number of TACACS operations than would be possible if multiple TCP connections were used to send requests to a server.

The TACACS+ server that is being used must support single-connection mode for this to be effective; otherwise, the connection between the network access server and the TACACS+ server locks up or you can receive unauthentic errors.

**Task ID**
```
<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>
```

**Examples**
The following example shows how to configure a single TCP connection to be made with the TACACS+ server (IP address 209.165.200.226) and all authentication, authorization, accounting requests to use this TCP connection. This works only if the TACACS+ server is also configured in single-connection mode. To configure the TACACS+ server in single connection mode, refer to the respective server manual.

```
RP/0/RP0/CPU0:router(config)# tacacs-server host 209.165.200.226
RP/0/RP0/CPU0:router(config-tacacs-host)# single-connection
```
tacacs-server host

To specify a TACACS+ host server, use the `tacacs-server host` command in XR Config mode. To delete the specified name or address, use the `no` form of this command.

```
tacacs-server host host-name [port port-number] [timeout seconds] [key [0 | 7] auth-key] [single-connection]
no tacacs-server host host-name [port port-number]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host-name</td>
<td>Host or domain name or IP address of the TACACS+ server.</td>
</tr>
<tr>
<td>port port-number</td>
<td>(Optional) Specifies a server port number. This option overrides the default, which is port 49. Valid port numbers range from 1 to 65535.</td>
</tr>
<tr>
<td>timeout seconds</td>
<td>(Optional) Specifies a timeout value that sets the length of time the authentication, authorization, and accounting (AAA) server waits to receive a response from the TACACS+ server. This option overrides the global timeout value set with the <code>tacacs-server timeout</code> command for this server only. The valid timeout range is from 1 to 1000 seconds. Default is 5.</td>
</tr>
<tr>
<td>key [0</td>
<td>7] auth-key</td>
</tr>
<tr>
<td>single-connection</td>
<td>(Optional) Multiplexes all TACACS+ requests to this server over a single TCP connection. By default, a separate connection is used for each session.</td>
</tr>
</tbody>
</table>

**Command Default**

No TACACS+ host is specified.

The `port-name` argument, if not specified, defaults to the standard port 49.

The `seconds` argument, if not specified, defaults to 5 seconds.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>
You can use multiple `tacacs-server host` commands to specify additional hosts. Cisco IOS XR software searches for hosts in the order in which you specify them.

### Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

### Examples

The following example shows how to specify a TACACS+ host with the IP address 209.165.200.226:

```
RP/0/RP0/CPU0:router(config)# tacacs-server host 209.165.200.226
```

The following example shows that the default values from the `tacacs-server host` command are displayed from the `show run` command:

```
RP/0/RP0/CPU0:router# show run
Building configuration...
!! Last configuration change at 13:51:56 UTC Mon Nov 14 2005 by lab
! tacacs-server host 209.165.200.226 port 49
  timeout 5
!
```

The following example shows how to specify that the router consult the TACACS+ server host named host1 on port number 51. The timeout value for requests on this connection is 30 seconds; the encryption key is a_secret.

```
RP/0/RP0/CPU0:router(config)# tacacs-server host host1 port 51
RP/0/RP0/CPU0:router(config-tacacs-host)# timeout 30
RP/0/RP0/CPU0:router(config-tacacs-host)# key a_secret
```
**tacacs-server key**

To set the authentication encryption key used for all TACACS+ communications between the router and the TACACS+ daemon, use the **tacacs-server key** command in XR Config mode. To disable the key, use the **no** form of this command.

```
tacacs-server key {0 clear-text-key | 7 encrypted-key auth-key}
no tacacs-server key {0 clear-text-key | 7 encrypted-key auth-key}
```

**Syntax Description**

- **0 clear-text-key**: Specifies an unencrypted (cleartext) shared key.
- **7 encrypted-key**: Specifies an encrypted shared key.
- **auth-key**: Specifies the unencrypted key between the AAA server and the TACACS+ server.

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The key name entered must match the key used on the TACACS+ daemon. The key name applies to all servers that have no individual keys specified. All leading spaces are ignored; spaces within and after the key are not. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

The key name is valid only when the following guidelines are followed:

- The **clear-text-key** argument must be followed by the **0** keyword.
- The **encrypted-key** argument must be followed by the **7** keyword.

The TACACS server key is used only if no key is configured for an individual TACACS server. Keys configured for an individual TACACS server always override this global key configuration.

**Examples**

The following example sets the authentication and encryption key to key1:

```
RP/0/RP0/CPU0:router(config)# tacacs-server key key1
```
tacacs-server timeout

To set the interval that the server waits for a server host to reply, use the `tacacs-server timeout` command in XR Config mode. To restore the default, use the `no` form of this command.

```
tacacs-server timeout seconds
no tacacs-server timeout seconds
```

**Syntax Description**

- `seconds` Integer that specifies the timeout interval (in seconds) from 1 to 1000.

**Command Default**

5 seconds

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The TACACS+ server timeout is used only if no timeout is configured for an individual TACACS+ server. Timeout intervals configured for an individual TACACS+ server always override this global timeout configuration.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows the interval timer being changed to 10 seconds:

```
RP/0/RP0/CPU0:router(config)# tacacs-server timeout 10
```
To set the Differentiated Services Code Point (DSCP), which is represented by the first six bits in the Type of Service (ToS) byte of the IP header, use the **tacacs-server ipv4** command in XR Config mode.

```
tacacs-server ipv4 dscp dscp-value
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ipv4</strong></td>
<td>Specifies the dscp bit for the IPv4 packets.</td>
</tr>
<tr>
<td><strong>dscp</strong></td>
<td>Sets the DSCP in the IP header.</td>
</tr>
<tr>
<td><strong>dscp-value</strong></td>
<td>Specifies the options for setting the value of DSCP. The available options are:</td>
</tr>
<tr>
<td></td>
<td>• &lt;0-63&gt; Differentiated services codepoint value</td>
</tr>
<tr>
<td></td>
<td>• af11 Match packets with AF11 dscp (001010)</td>
</tr>
<tr>
<td></td>
<td>• af12 Match packets with AF12 dscp (001100)</td>
</tr>
<tr>
<td></td>
<td>• af13 Match packets with AF13 dscp (001110)</td>
</tr>
<tr>
<td></td>
<td>• af21 Match packets with AF21 dscp (010010)</td>
</tr>
<tr>
<td></td>
<td>• af22 Match packets with AF22 dscp (010100)</td>
</tr>
<tr>
<td></td>
<td>• af23 Match packets with AF23 dscp (010110)</td>
</tr>
<tr>
<td></td>
<td>• af31 Match packets with AF31 dscp (011010)</td>
</tr>
<tr>
<td></td>
<td>• af32 Match packets with AF32 dscp (011100)</td>
</tr>
<tr>
<td></td>
<td>• af33 Match packets with AF33 dscp (011110)</td>
</tr>
<tr>
<td></td>
<td>• af41 Match packets with AF41 dscp (100010)</td>
</tr>
<tr>
<td></td>
<td>• af42 Match packets with AF42 dscp (100100)</td>
</tr>
<tr>
<td></td>
<td>• af43 Match packets with AF43 dscp (100110)</td>
</tr>
<tr>
<td></td>
<td>• cs1 Match packets with CS1(precedence 1) dscp (001000)</td>
</tr>
<tr>
<td></td>
<td>• cs2 Match packets with CS2(precedence 2) dscp (010000)</td>
</tr>
<tr>
<td></td>
<td>• cs3 Match packets with CS3(precedence 3) dscp (011000)</td>
</tr>
<tr>
<td></td>
<td>• cs4 Match packets with CS4(precedence 4) dscp (100000)</td>
</tr>
<tr>
<td></td>
<td>• cs5 Match packets with CS5(precedence 5) dscp (101000)</td>
</tr>
<tr>
<td></td>
<td>• cs6 Match packets with CS6(precedence 6) dscp (110000)</td>
</tr>
<tr>
<td></td>
<td>• cs7 Match packets with CS7(precedence 7) dscp (111000)</td>
</tr>
<tr>
<td></td>
<td>• default Match packets with default dscp (000000)</td>
</tr>
<tr>
<td></td>
<td>• ef Match packets with EF dscp (101110)</td>
</tr>
</tbody>
</table>
**tacacs-server ipv4**

**Command Default**
None

**Command Modes**
XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
No specific guidelines impact the use of this command.

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**
The following example sets the DSCP value to Assured Forwarding (AF)11:
```
RP/0/RP0/CPU0:router(config)# tacacs-server ipv4 dscp af11
```
tacacs source-interface

To specify the source IP address of a selected interface for all outgoing TACACS+ packets, use the tacacs source-interface command in XR Config mode. To disable use of the specified interface IP address, use the no form of this command.

`tacacs source-interface type path-id [vrf vrf-id]
no tacacs source-interface type path-id`

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>type</code></td>
<td>Interface type. For more information, use the question mark (?) online help function.</td>
</tr>
<tr>
<td><code>path-id</code></td>
<td>Physical interface or virtual interface.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Use the show interfaces command in XR Config mode to see a list of all interfaces currently configured on the router.</td>
</tr>
<tr>
<td><code>vrf vrf-id</code></td>
<td>Specifies the name of the assigned VRF.</td>
</tr>
</tbody>
</table>

**Command Default**

If a specific source interface is not configured, or the interface is down or does not have an IP address configured, the system selects an IP address.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the tacacs source-interface command to set the IP address of the specified interface for all outgoing TACACS+ packets. This address is used as long as the interface is in the up state. In this way, the TACACS+ server can use one IP address entry associated with the network access client instead of maintaining a list of all IP addresses.

This command is especially useful in cases where the router has many interfaces and you want to ensure that all TACACS+ packets from a particular router have the same IP address.

When the specified interface does not have an IP address or is in a down state, TACACS+ behaves as if no source interface configuration is used.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to set the IP address of the specified interface for all outgoing TACACS+ packets:
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# tacacs source-interface HundredGigabitEthernet 0/0/0/29 vrf abc
**task**

To add a task ID to a task group, use the **task** command in task group configuration mode. To remove a task ID from a task group, use the **no** form of this command.

```
task {read | write | execute | debug} taskid-name
no task {read | write | execute | debug} taskid-name
```

**Syntax Description**

- **read** Enables read-only privileges for the named task ID.
- **write** Enables write privileges for the named task ID. The term “write” implies read also.
- **execute** Enables execute privileges for the named task ID.
- **debug** Enables debug privileges for the named task ID.

**taskid-name** Name of the task ID.

**Command Default**

No task IDs are assigned to a newly created task group.

**Command Modes**

Task group configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the **task** command in task group configuration mode. To access task group configuration mode, use the **taskgroup** command in global configuration mode.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to enable execute privileges for the config-services task ID and associate that task ID with the task group named taskgroup1:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# taskgroup taskgroup1
RP/0/RP0/CPU0:router(config-tg)# task execute config-services
```
To configure a task group to be associated with a set of task IDs, and to enter task group configuration mode, use the `taskgroup` command in XR Config mode. To delete a task group, use the `no` form of this command.

```
taskgroup taskgroup-name [{description string | task {read | write | execute | debug} taskid-name | inherit taskgroup taskgroup-name}]

no taskgroup taskgroup-name
```

**Syntax Description**
- `taskgroup-name`: Name of a particular task group.
- `description`: (Optional) Enables you to create a description for the named task group.
- `string`: (Optional) Character string used for the task group description.
- `task`: (Optional) Specifies that a task ID is to be associated with the named task group.
- `read`: (Optional) Specifies that the named task ID permits read access only.
- `write`: (Optional) Specifies that the named task ID permits read and write access only.
- `execute`: (Optional) Specifies that the named task ID permits execute access.
- `debug`: (Optional) Specifies that the named task ID permits debug access only.
- `taskid-name`: (Optional) Name of a task: the task ID.
- `inherit taskgroup`: (Optional) Copies permissions from the named task group.
- `taskgroup-name`: (Optional) Name of the task group from which permissions are to be inherited.

**Command Default**
Five predefined user groups are available by default.

**Command Modes**
XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
Task groups are configured with a set of task IDs for each action type. Deleting a task group that is still referenced in the system results in a warning and rejection of the deletion.

From global configuration mode, you can display all the configured task groups. However, you cannot display all the configured task groups in task group configuration mode.

Entering the `taskgroup` command with no keywords or arguments enters task group configuration mode, in which you can use the `description`, `inherit`, `show`, and `task` commands.
<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td></td>
<td>read, write</td>
</tr>
</tbody>
</table>

### Examples

The following example assigns read bgp permission to the task group named alpha:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# taskgroup alpha
RP/0/RP0/CPU0:router(config-tg)# task read bgp
```
timeout (TACACS+)

To specify a timeout value that sets the length of time the authentication, authorization, and accounting (AAA) server waits to receive a response from the TACACS+ server, use the `timeout` (TACACS+) command in TACACS host configuration mode. To disable this command and return to the default timeout value of 5 seconds, use the `no` form of this command.

```
timeout seconds
no timeout seconds
```

**Syntax Description**

- `seconds` Timeout value (in seconds). The range is from 1 to 1000. If no timeout is specified, the global value is used.

**Command Default**

`seconds`: 5

**Command Modes**

TACACS host configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `timeout` (TACACS+) command overrides the global timeout value set with the `tacacs-server timeout` command for this server only.

**Task ID**

- `aaa` read, write

**Examples**

The following example shows how to set the number of seconds for the timeout value:

```
RP/0/RP0/CPU0:router(config)# tacacs-server host 209.165.200.226
RP/0/RP0/CPU0:router(config-tacacs-host)# timeout 500
```
timeout login response

To set the interval that the server waits for a reply to a login, use the `timeout login response` command in line template configuration mode. To restore the default, use the `no` form of this command.

```
timeout login response seconds
no timeout login response seconds
```

**Syntax Description**

`seconds` Integer that specifies the timeout interval (in seconds) from 0 to 300.

**Command Default**

`seconds`: 30

**Command Modes**

Line template configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `timeout login response` command in line template configuration mode to set the timeout value. This timeout value applies to all terminal lines to which the entered line template is applied. This timeout value cannot be applied to line console. After the timeout value has expired, the user is prompted again. The retry is allowed three times.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to change the interval timer to 20 seconds:

```
RP/0/RP0/CP00:router# configure
RP/0/RP0/CP00:router(config)# line template alpha
RP/0/RP0/CP00:router(config-line)# timeout login response 20
```
To configure a user group and associate it with a set of task groups, and to enter user group configuration mode, use the `usergroup` command in XR Config mode. To delete a user group, or to delete a task-group association with the specified user group, use the `no` form of this command.

**Syntax Description**

```plaintext
usergroup  usergroup-name
no  usergroup  usergroup-name
```

- `usergroup-name`: Name of the user group. The `usergroup-name` argument can be only one word. Spaces and quotation marks are not allowed.

**Command Default**

Five predefined user groups are available by default.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

User groups are configured with the command parameters for a set of users, such as task groups. You can remove specific user groups by using the `no` form of the `usergroup` command. You can remove the user group itself by using the `no` form of the command without giving any parameters. Deleting a user group that is still referenced in the system results in a warning and a rejection of the deletion.

Use the `inherit usergroup`, on page 28 command to copy permissions from other user groups. The user group is inherited by the parent group and forms a union of all task IDs specified in those groups. Circular inclusions are detected and rejected. User groups cannot inherit properties from predefined groups, such as root-system and owner-sdr.

From global configuration mode, you can display all the configured user groups. However, you cannot display all the configured user groups in usergroup configuration mode.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to add permissions from the user group beta to the user group alpha:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# usergroup alpha
RP/0/RP0/CPU0:router(config-ug)# inherit usergroup beta
```
**username**

To configure a new user with a username, establish a password, associate a password policy with the user, grant permissions for the user, and to enter username configuration mode, use the `username` command in XR Config mode or System Admin Config mode. To delete a user from the database, use the `no` form of this command.

```
username user-name [{password {0 | 7} password | policy common-policy-name | secret {0 | 5 | 8 | 9 | 10} password | group usergroup-name}]

no username user-name [{password {0 | 7} password | policy common-policy-name | secret {0 | 5 | 8 | 9 | 10} password | group usergroup-name}]
```

### Syntax Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user-name</td>
<td>Name of the user. The <code>user-name</code> argument can be only one word. Spaces and quotation marks are not allowed. The allowed range for a user-defined username is 2-253 characters.</td>
</tr>
<tr>
<td>password</td>
<td>(Optional) Enables a password to be created for the named user.</td>
</tr>
<tr>
<td>0</td>
<td>(Optional) Specifies that an unencrypted (clear-text) password follows. The password will be encrypted for storage in the configuration using a Cisco proprietary encryption algorithm.</td>
</tr>
<tr>
<td>7</td>
<td>(Optional) Specifies that an encrypted password follows.</td>
</tr>
<tr>
<td>password</td>
<td>(Optional) Specifies the unencrypted password text to be entered by the user to log in, for example, “lab”. If encryption is configured, the password is not visible to the user. Can be up to 253 characters in length.</td>
</tr>
<tr>
<td>policy</td>
<td>Configures the policy that is common to both user password and secret.</td>
</tr>
<tr>
<td>common-policy-name</td>
<td>Specifies the name of the common password policy.</td>
</tr>
</tbody>
</table>
**username**

<table>
<thead>
<tr>
<th>secret</th>
<th>(Optional) Enables an MD5-secured password to be created for the named user.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(Optional) Specifies that an unencrypted (clear-text) password follows. The password will be encrypted for storage in the configuration using an MD5 encryption algorithm.</td>
</tr>
<tr>
<td>5</td>
<td>(Optional) Specifies that an encrypted password follows.</td>
</tr>
<tr>
<td>8</td>
<td>(Optional) Specifies that Type 8 password that uses SHA256 hashing algorithm follows.</td>
</tr>
<tr>
<td>9</td>
<td>(Optional) Specifies that Type 9 password that uses scrypt hashing algorithm follows.</td>
</tr>
<tr>
<td>10</td>
<td>(Optional) Specifies that Type 10 password that uses SHA512 hashing algorithm follows.</td>
</tr>
<tr>
<td>group</td>
<td>(Optional) Enables a named user to be associated with a user group.</td>
</tr>
<tr>
<td>usergroup-name</td>
<td>(Optional) Name of a user group as defined with the <strong>usergroup</strong> command.</td>
</tr>
</tbody>
</table>

**Command Default**
No usernames are defined in the system.

**Command Modes**
- XR Config mode
- System Admin Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>Release 7.0.1</td>
<td>Added the support for Type 8 (SHA256), Type 9 (scrypt) and Type 10 (SHA512) for <strong>secret</strong> configuration.</td>
</tr>
<tr>
<td>Release 7.2.1</td>
<td>Added the support for <strong>policy</strong> option to configure policy common to user password and secret.</td>
</tr>
</tbody>
</table>
Usage Guidelines

- A user is never allowed to have cisco-support privileges as the only group.
- From Release 7.0.1 and later, Type 10 (SHA512) is applied as the default type for the secret configuration. Prior to this, Type 5 (MD5) was the default one.

Use the `username` command to identify the user and enter username configuration mode. Password and user group assignments can be made from either XR Config mode or username configuration submode. Permissions (task IDs) are assigned by associating the user with one or more defined user groups.

From XR Config mode, you can display all the configured usernames. You can display configured usernames in configuration mode by `router(config): do show run username`.

Each user is identified by a username that is unique across the administrative domain. Each user should be made a member of at least one user group. Deleting a user group may orphan the users associated with that group. The AAA server authenticates orphaned users, but most commands are not authorized.

The `username` command is associated with a particular user for local login authentication by default. Alternatively, a user and password can be configured in the database of the TACACS+ server for TACACS+ login authentication. For more information, see the description of the `aaa authentication (XR-VM)`, on page 9 command.

The predefined group root-system may be specified only by root-system users while administration is configured.

To enable the local networking device to respond to remote Challenge Handshake Authentication Protocol (CHAP) challenges, one `username` command entry must be the same as the hostname entry that has already been assigned to the other networking device.

```
RP/0/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# username user1
RP/0/RP0/CPU0:router(config-un)# ?
```

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

The following example shows the commands available after executing the `username` command:

```
clear    Clear the uncommitted configuration
commit   Commit the configuration changes to running
describe Describe a command without taking real actions
do      Run an exec command
```
The following examples show how to establish the clear-text password `password1` for the user name `user1`:

```
RP/0/RP0/CPU0:router(config-un)#

The following example shows how to establish the clear-text password `password1` for the user name `user1`:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# username user1
RP/0/RP0/CPU0:router(config-un)# password 0 password1
```

This example shows how to apply a password policy for the user secret:

```
Router# configure
Router(config)# username user1
Router(config-un)# policy test-policy1
Router(config-un)# secret 10
$6$dmwuW0Ajicf9Bw0.Sy/vzynWFl/0cGaq4hMh8279VHy5ZZLhoHd7TicP+4mOo8IIrYCGA90U.A.w1JYTP7lBzrY.DxHrE3SN28BzB3e0
Router(config-un)# commit
```

The following example shows how to configure a Type 8 (SHA256) password for the user, `user8`. You can also see the examples and usage of the `secret`, on page 42 command.

You can specify Type as '8' under the `secret` keyword, to explicitly configure Type 8 password.

```
Router# configure
Router(config)# username user8 secret 8
$8$ZYKGl1dZIw73Dl$SUWJOqTLoMyExhs1NKoL5vMtvCOYguM5ajXf4uGeQj6I
Router(config-un)# commit
```

This example shows how to configure Type 9 password:

```
Router# configure
Router(config)# username user9 secret 9
$9/$rIQL183r1P4L/$0sZfJfWKFyH6B/kAxpkkXmIqPjRkFPEoh3WgGbwQ
Router(config-un)# commit
```

Similarly, this example shows how to configure Type 10 password:

```
Router# configure
Router(config)# username user10 secret 10
$6$9우J5dvegetEFqgKAPJ53CLIL1/F.E4v/Hi.UEgQwK8UkS6kR9P9G6c55Q9t2gM34jC8BQ?meAwyhu5VM/aRFfJgk/jxZG17h6xPrvJWf1
Router(config-un)# commit
```
This example shows how to specify the Type 10 password in System Admin VM:

Router#admin
sysadmin-vm:0_RP0# configure
sysadmin-vm:0_RP0(config)# aaa authentication users user user10 password testpassword
sysadmin-vm:0_RP0(config)# commit
users group

To associate a user group and its privileges with a line, use the `users group` command in line template configuration mode. To delete a user group association with a line, use the `no` form of this command.

```
users group {usergroup-name | cisco-support | maintenance | netadmin | operator | provisioning | retrieve | root-lr | serviceadmin | sysadmin}
```

```
o users group {usergroup-name | cisco-support | maintenance | netadmin | operator | provisioning | retrieve | root-lr | serviceadmin | sysadmin}
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usergroup-name</td>
<td>Name of the user group. The <code>usergroup-name</code> argument can be only one word. Spaces and quotation marks are not allowed.</td>
</tr>
<tr>
<td>cisco-support</td>
<td>Specifies that users logging in through the line are given Cisco support personnel privileges.</td>
</tr>
<tr>
<td>maintenance</td>
<td>Specifies that users logging in through the line are given SCAPA maintenance privileges.</td>
</tr>
<tr>
<td>netadmin</td>
<td>Specifies that users logging in through the line are given network administrator privileges.</td>
</tr>
<tr>
<td>operator</td>
<td>Specifies that users logging in through the line are given operator privileges.</td>
</tr>
<tr>
<td>provisioning</td>
<td>Specifies that users logging in through the line are given SCAPA provisioning privileges.</td>
</tr>
<tr>
<td>retrieve</td>
<td>Specifies that users logging in through the line are given SCAPA retrieve privileges.</td>
</tr>
<tr>
<td>root-lr</td>
<td>Specifies that users logging in through the line are given root logical router (LR) privileges.</td>
</tr>
<tr>
<td>serviceadmin</td>
<td>Specifies that users logging in through the line are given service administrator group privileges.</td>
</tr>
<tr>
<td>sysadmin</td>
<td>Specifies that users logging in through the line are given system administrator privileges.</td>
</tr>
</tbody>
</table>
Command Default
None

Command Modes
Line template configuration

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines
Use the `users group` command to enable a user group and its privileges to be associated with a line, meaning that users logging in through the line are given the privileges of the particular user group.

Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

Examples
In the following example, if a vty-pool is created with line template `vty`, users logging in through `vty` are given operator privileges:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa authen login vty-authen line
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router(config)# line template vty
RP/0/RP0/CPU0:router(config-line)# users group operator
RP/0/RP0/CPU0:router(config-line)# login authentication
```
vrf (RADIUS)

To configure the Virtual Private Network (VPN) routing and forwarding (VRF) reference of an AAA RADIUS server group, use the `vrf` command in RADIUS server-group configuration mode. To enable server groups to use the global (default) routing table, use the `no` form of this command.

```
vrf vrf-name
no vrf vrf-name
```

**Syntax Description**

- `vrf-name` Name assigned to a VRF.

**Command Default**

The default VRF is used.

**Command Modes**

RADIUS server-group configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `vrf` command to specify a VRF for an AAA RADIUS server group and enable dial-up users to use AAA servers in different routing domains.

**Examples**

The following example shows how to use the `vrf` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa group server radius group1
RP/0/RP0/CPU0:router(config-sg-radius)# vrf vrf1
```
vrf (TACACS+)

To configure the Virtual Private Network (VPN) routing and forwarding (VRF) reference of an AAA TACACS+ server group, use the vrf command in TACACS+ server-group configuration mode. To enable server groups to use the global (default) routing table, use the no form of this command.

```
vrf vrf-name
no vrf vrf-name
```

**Syntax Description**
- `vrf-name` Name assigned to a VRF.

**Command Default**
The default VRF is used.

**Command Modes**
TACACS+ server-group configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
Use the vrf command to specify a VRF for an AAA TACACS+ server group and enable dial-up users to use AAA servers in different routing domains.

**Task ID**
```
Task ID  Operations
aaa      read, write
```

**Examples**
This example shows how to use the vrf command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa group server tacacs+ myserver
RP/0/RP0/CPU0:router(config-sg-tacacs+)# server 9.27.10.6
RP/0/RP0/CPU0:router(config-sg-tacacs+)# vrf abc
```
Keychain Management Commands

This module describes the commands used to configure keychain management.

All commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router that is introduced from Cisco IOS XR Release 6.3.2. References to earlier releases in Command History tables apply to only the Cisco NCS 5500 Series Router.

Note

• Starting with Cisco IOS XR Release 6.6.25, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 560 Series Routers.

• Starting with Cisco IOS XR Release 6.3.2, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router.

• References to releases before Cisco IOS XR Release 6.3.2 apply to only the Cisco NCS 5500 Series Router.

• Cisco IOS XR Software Release 7.0.1 specific updates are not applicable for the following variants of Cisco NCS 540 Series Routers:
  • N540-28Z4C-SYS-A
  • N540-28Z4C-SYS-D
  • N540X-16Z4G8Q2C-A
  • N540X-16Z4G8Q2C-D
  • N540-12Z20G-SYS-A
  • N540-12Z20G-SYS-D
  • N540X-12Z16G-SYS-A
  • N540X-12Z16G-SYS-D
For detailed information about keychain management concepts, configuration tasks, and examples, see the Implementing Keychain Management chapter in the *System Security Configuration Guide for Cisco NCS 5500 Series Routers*.

**Note**

Currently, only default VRF is supported. VPNv4, VPNv6 and VPN routing and forwarding (VRF) address families will be supported in a future release.

- accept-lifetime, on page 107
- accept-tolerance, on page 108
- ao, on page 109
- cryptographic-algorithm, on page 110
- key (key chain), on page 112
- key (tcp ao keychain), on page 113
- keychain, on page 114
- tcp ao, on page 115
- key chain (key chain), on page 116
- key config-key password-encryption, on page 117
- key-string (keychain), on page 118
- send-lifetime, on page 120
- show key chain, on page 121
- show type6, on page 122
**accept-lifetime**

To set the time period during which the authentication key on a keychain is received as valid, use the `accept-lifetime` command in key configuration mode. To revert to the default value, use the `no` form of this command.

```
accept-lifetime  start-time  [[duration duration value | infinite|end-time]]
no  accept-lifetime  start-time  [[duration duration value | infinite|end-time]]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>start-time</code></td>
<td>Start time, in <code>hh:mm:ss day month year</code> format, in which the key becomes valid.</td>
</tr>
<tr>
<td></td>
<td>The range is from 0:0:0 to 23:59:59.</td>
</tr>
<tr>
<td></td>
<td>The range for the number of days of the month is from 1 to 31.</td>
</tr>
<tr>
<td></td>
<td>The range for the years is from 1993 to 2035.</td>
</tr>
<tr>
<td><code>duration</code></td>
<td>(Optional) Determines the lifetime of the key in seconds. The range is from 1-2147483646.</td>
</tr>
<tr>
<td>value</td>
<td>(Optional) Specifiesthatthekeyneverexpiresafteritbecomesvalid.</td>
</tr>
<tr>
<td><code>infinite</code></td>
<td>(Optional) Specifies that the key never expires after it becomes valid.</td>
</tr>
<tr>
<td><code>end-time</code></td>
<td>(Optional) Time, in <code>hh:mm:ss day month year</code> format, after which the key expires.</td>
</tr>
<tr>
<td></td>
<td>The range is from 0:0:0 to 23:59:59.</td>
</tr>
</tbody>
</table>

**Command Default**

None

**Command Modes**

Key configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to use the `accept-lifetime` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0/CPU0:router(config-isis-keys)# key 8
RP/0/RP0/CPU0:router(config-isis-keys-0x8)# accept-lifetime 1:00:00 June 29 2006 infinite
```
accept-tolerance

To specify the tolerance or acceptance limit, in seconds, for an accept key that is used by a peer, use the accept-tolerance command in keychain configuration mode. To disable this feature, use the no form of this command.

```
accept-tolerance [{value | infinite}]
no accept-tolerance [{value | infinite}]
```

### Syntax Description

- **value** *(Optional)* Tolerance range, in seconds. The range is from 1 to 8640000.
- **infinite** *(Optional)* Specifies that the tolerance specification is infinite. The accept key never expires. The tolerance limit of infinite indicates that an accept key is always acceptable and validated when used by a peer.

### Command Default

The default value is 0, which is no tolerance.

### Command Modes

Keychain configuration

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

If you do not configure the accept-tolerance command, the tolerance value is set to zero.

Even though the key is outside the active lifetime, the key is deemed acceptable as long as it is within the tolerance limit (for example, either prior to the start of the lifetime, or after the end of the lifetime).

### Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

### Examples

The following example shows how to use the accept-tolerance command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0/CPU0:router(config-isis-keys)# accept-tolerance infinite
```
To specify the name the key chain used in the authentication option \texttt{ao} command in BGP neighbor configuration mode.

\texttt{ao key-chain-name \{inheritance-disable | include-tcp-options \{disable | enable \} accept-ao-mismatch-connection\}}

### Syntax Description

- **key-chain-name**: Specifies the name of the key chain. String of maximum length of 32 characters.
- **inheritance-disable**: Prevents the key chain from being inherited from the parent.
- **include-tcp-options**: Includes or excludes other TCP options in the header for MAC calculation.
  - **disable**: Excludes other TCP options in the header.
  - **enable**: Includes other TCP options in the header.
- **accept-ao-mismatch-connection**: Accepts connection even if there is a mismatch of AO options between peers.

### Command Default

The key chain has no specified name.

### Command Modes

BGP neighbor

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

This example shows how to specify the name the key chain used in the authentication option:

\begin{verbatim}
RP/0/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router(config)#router bgp 100
RP/0/RP0/CPU0:router(config-bgp)#neighbor 10.51.51.1
RP/0/RP0/CPU0:router(config-bgp-nbr)#address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr)#ao tcpao1 include-tcp-options disable accept-ao-mismatch-connection
\end{verbatim}
**cryptographic-algorithm**

To apply the cryptographic algorithm to the packets using the key string configured for the key ID, use the `cryptographic-algorithm` command in keychain-key configuration mode. To disable this feature, use the `no` form of this command.

```
cryptographic-algorithm [{ HMAC-MD5 | HMAC-SHA1-12 | HMAC-SHA1-20 | MD5 | SHA-1 | HMAC-SHA-256 | HMAC-SHA1-96 | AES-128-CMAC-96 }]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMAC-MD5</td>
<td>Configures HMAC-MD5 as a cryptographic algorithm with a digest size of 16 bytes.</td>
</tr>
<tr>
<td>HMAC-SHA1-12</td>
<td>Configures HMAC-SHA1-12 as a cryptographic algorithm with a digest size of 12 bytes.</td>
</tr>
<tr>
<td>HMAC-SHA1-20</td>
<td>Configures HMAC-SHA1-20 as a cryptographic algorithm with a digest size of 20 bytes.</td>
</tr>
<tr>
<td>MD5</td>
<td>Configures MD5 as a cryptographic algorithm with a digest size of 16 bytes.</td>
</tr>
<tr>
<td>SHA-1</td>
<td>Configures SHA-1-20 as a cryptographic algorithm with a digest size of 20 bytes.</td>
</tr>
<tr>
<td>HMAC-SHA-256</td>
<td>Configures HMAC-SHA-256 as a cryptographic algorithm with a digest size of 32 bytes.</td>
</tr>
<tr>
<td>HMAC-SHA1-96</td>
<td>Configures HMAC-SHA1-96 as a cryptographic algorithm with a digest size of 12 bytes.</td>
</tr>
<tr>
<td>AES-128-CMAC-96</td>
<td>Configures AES-128-CMAC as a cryptographic algorithm with a digest size of 12 bytes.</td>
</tr>
</tbody>
</table>

**Command Default**

No default behavior or values

**Command Modes**

Keychain-key configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>Release 6.5.1</td>
<td>Support for the following algorithms are added:</td>
</tr>
<tr>
<td></td>
<td>• HMAC-SHA-256</td>
</tr>
<tr>
<td></td>
<td>• HMAC-SHA1-96</td>
</tr>
<tr>
<td></td>
<td>• AES-128-CMAC-96</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

If you do not specify the cryptographic algorithm, MAC computation and API verification would be invalid.

These protocols support the following cryptographic algorithms:

- Border Gateway Protocol (BGP) supports only HMAC-MD5, HMAC-SHA1-12, AES-128-CMAC-96 and HMAC-SHA1-96.

• Open Shortest Path First (OSPF) supports MD5, HMAC-MD5, HMAC-SHA-256, HMAC-SHA1-12, HMAC-SHA1-20, and HMAC-SHA1-96.

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>system</td>
<td>read,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>write</td>
</tr>
</tbody>
</table>

Examples

The following example shows how to use the `cryptographic-algorithm` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0/CPU0:router(config-isis-keys)# key 8
RP/0/RP0/CPU0:router(config-isis-keys=0x8)# cryptographic-algorithm HMAC-MD5
```
key (key chain)

To create or modify a keychain key, use the key command in keychain-key configuration mode. To disable this feature, use the no form of this command.

```
key  key-id
no key  key-id
```

**Syntax Description**

- `key-id` 48-bit integer key identifier of from 0 to 281474976710655.

**Command Default**

No default behavior or values

**Command Modes**

Keychain-key configuration

**Command History**

- **Release 6.0**  This command was introduced.

**Usage Guidelines**

For a Border Gateway Protocol (BGP) keychain configuration, the range for the `key-id` argument must be from 0 to 63. If the range is above the value of 63, the BGP keychain operation is rejected.

**Task ID**

```
Task ID  Operations
        system  read, write
```

**Examples**

The following example shows how to use the key command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0/CPU0:router(config-isis-keys)# key 8
RP/0/RP0/CPU0:router(config-isis-keys-0x8)#
```
key (tcp ao keychain)

To configure in send and receive identifiers for the key, use the key command in TCP authentication option keychain configuration mode.

```
key key-identifier sendID send-id-value ReceiveID receive-id-value
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key-identifier</td>
<td>Identifier of the key. Acceptable values are 48-bit integers. Range is 0 to 2814796710655.</td>
</tr>
<tr>
<td>SendID send-id-value</td>
<td>Specifies the send identifier value. Range is 0 to 255.</td>
</tr>
<tr>
<td>ReceiveID receive-id-value</td>
<td>Specifies the receive identifier value to be used for the key. The range is 0 to 255.</td>
</tr>
</tbody>
</table>

**Command Default**
The key is not enabled.

**Command Modes**
TCP authentication option keychain

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>bgp</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**

This example shows how to configure the send and receive identifier for the key.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# tcp ao
RP/0/RP0/CPU0:router(conf-tcp-ao)# keychain tcpao1
RP/0/RP0/CPU0:router(config-tcp-ao-tcpao1)# key 10 sendID 5 receiveID 5
```
keychain

To configure the keychain to be used in TCP authentication option, use the `tcp ao` command in TCP authentication option configuration mode.

```
keychain  keychain-name
```

**Syntax Description**
This command has no arguments or keywords.

**Command Default**
The keychain is not enabled.

**Command Modes**
TCP authentication option

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>bgp</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**
This example shows how to configure the `keychain` for TCP Authentication option:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# tcp ao
RP/0/RP0/CPU0:router(config-tcp-ao)# keychain tcpao1
```
**tcp ao**

To enable the TCP authentication option, use the `tcp ao` command in global configuration mode.

```
tcp ao
no tcp ao
```

**Syntax Description**
This command has no arguments or keywords.

**Command Default**
The TCP authentication option is not enabled.

**Command Modes**
Global configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>bgp</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**

This example shows how to configure the `tcp ao` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# tcp ao
```
key chain (key chain)

To create or modify a keychain, use the `key chain` command. To disable this feature, use the `no` form of this command.

```
key chain key-chain-name
no key chain key-chain-name
```

**Syntax Description**

- `key-chain-name` Specifies the name of the keychain. The maximum number of characters is 48.

**Command Default**

No default behavior or values

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

You can configure a keychain for Border Gateway Protocol (BGP) as a neighbor, session group, or neighbor group. BGP can use the keychain to implement a hitless key rollover for authentication.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows that the name of the keychain `isis-keys` is for the `key chain` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0/CPU0:router(config-isis-keys)#
```
**key config-key password-encryption**

To create a primary key for the Type 6 password encryption feature, use the `key config-key password-encryption` command in the EXEC mode.

```
key config-key password-encryption  [delete]
```

**Syntax Description**
- `delete` (Optional) Deletes the primary key for Type 6 password encryption.

**Command Default**
No primary key exists.

**Command Modes**
EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to create a primary key for Type 6 password encryption:

```
Router# key config-key password-encryption
New password Requirements: Min-length 6, Max-length 64
Characters restricted to [A-Z][a-z][0-9]
Enter new key :
Enter confirm key :
Master key operation is started in background
```

The following example shows how to delete a primary key for Type 6 password encryption:

```
Router# key config-key password-encryption delete
WARNING: All type 6 encrypted keys will become unusable
Continue with master key deletion ? [yes/no]: yes
Master key operation is started in background
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>password6 encryption aes</td>
<td>Enables Type 6 password encryption feature.</td>
</tr>
<tr>
<td>show type6 server</td>
<td>Displays Type 6 password information.</td>
</tr>
</tbody>
</table>
key-string (keychain)

To specify the text string for the key, use the key-string command in keychain-key configuration mode. To disable this feature, use the no form of this command.

```
key-string  [{clear | password}]  key-string-text
no  key-string  [{clear | password}]  key-string-text
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>clear</th>
<th>Specifies the key string in clear-text form.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>password</td>
<td>Specifies the key in encrypted form.</td>
</tr>
<tr>
<td>key-string-text</td>
<td></td>
<td>Text string for the key, which is encrypted by the parser process before being saved to the configuration. The text string has the following character limitations:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Plain-text key strings—Minimum of 1 character and a maximum of 32.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Encrypted key strings—Minimum of 4 characters and no maximum.</td>
</tr>
</tbody>
</table>

Command Default

The default value is clear.

Command Modes

Keychain-key configuration

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines

For an encrypted password to be valid, the following statements must be true:

• String must contain an even number of characters, with a minimum of four.

• The first two characters in the password string must be decimal numbers and the rest must be hexadecimals.

• The first two digits must not be a number greater than 53.

Either of the following examples would be valid encrypted passwords:

1234abcd

or

50aefd

From Cisco IOS XR Software Release 7.1.2, Release 7.2.1 and later, if you are using any HMAC-SHA algorithm for a session, then you must ensure that the configured key-string has a minimum length of 14 characters. Otherwise, the session goes down. This guideline is applicable only for FIPS mode.
The following example shows how to use the `keystring` command:

```
RP/0/RP0/CPU0:router(# configure
RP/0/RP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0/CPU0:router(config-isis-keys)# key 8
RP/0/RP0/CPU0:router(config-isis-keys-0x8)# key-string password 850aefd
```
send-lifetime

To send the valid key and to authenticate information from the local host to the peer, use the send-lifetime command in keychain-key configuration mode. To disable this feature, use the no form of this command.

\[
\text{send-lifetime} \ start-time \ [(\text{duration} \ duration \ value | \text{infinite} \ end-time)] \\
\text{no} \ \text{send-lifetime} \ start-time \ [(\text{duration} \ duration \ value | \text{infinite} \ end-time)]
\]

**Syntax Description**

- **start-time**: Start time, in `hh:mm:ss day month year` format, in which the key becomes valid. The range is from 0:0:0 to 23:59:59. The range for the number of days of the month to start is from 1 to 31. The range for the years is from 1993 to 2035.

- **duration duration value**: (Optional) Determines the lifetime of the key in seconds.

- **infinite**: (Optional) Specifies that the key never expires once it becomes valid.

- **end-time**: (Optional) Time, in `hh:mm:ss day month year` format, after which the key expires. The range is from 0:0:0 to 23:59:59.

**Command Default**

No default behavior or values

**Command Modes**

Keychain-key configuration

**Command History**

- **Release 6.0**: This command was introduced.

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

- **Task ID**: system read, write

**Examples**

The following example shows how to use the `send-lifetime` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0/CPU0:router(config-isis-keys)# key 8
RP/0/RP0/CPU0:router(config-isis-keys-0x8)# send-lifetime 1:00:00 June 29 2006 infinite
```
show key chain

To display the keychain, use the show key chain command.

```
show key chain key-chain-name
```

**Syntax Description**

*key-chain-name*  Names of the keys in the specified keychain. The maximum number of characters is 32.

**Command Default**

If the command is used without any parameters, then it lists out all the key chains.

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**

When a secure key storage becomes available, it is desirable for keychain management to alternatively prompt you for a primary password and display the key label after decryption. The following example displays only the encrypted key label for the show key chain command:

```
RP/0/RP0/CPU0:router# show key chain isis-keys

Key-chain: isis-keys/ -
accept-tolerance -- infinite
Key 8 -- text "8"
cryptographic-algorithm -- MD5
Send lifetime: 01:00:00, 29 Jun 2006 - Always valid [Valid now]
Accept lifetime: 01:00:00, 29 Jun 2006 - Always valid [Valid now]
```
show type6

To view Type 6 password encryption information, use the `show type6` command in EXEC mode.

```
show type6 {clients | server | trace server} {all | error | info} [trace-server-parameter]
```

**Syntax Description**

- **clients** Displays Type 6 client information.
- **server** Displays Type 6 server information.
- **trace server** Displays Type 6 trace server information.
- **all** Displays all Type 6 traces.
- **error** Displays Type 6 error traces.
- **info** Displays Type 6 information trace entries.

**trace-server-parameter** (Optional) Displays Type 6 trace server information for the specified parameter. Use one from the list of parameters defined in the Usage Guidelines section.

**Command Default**

None.

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

In the command form `show type6 trace server info trace-server-parameter`, replace `trace-server-parameter` with one of the following parameters:

<table>
<thead>
<tr>
<th>Trace Server Parameter</th>
<th>Displayed Trace Server Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>The specified file.</td>
</tr>
<tr>
<td>hexdump</td>
<td>Hexadecimal format.</td>
</tr>
<tr>
<td>last</td>
<td>The most recent entries.</td>
</tr>
<tr>
<td>location</td>
<td>Line card location.</td>
</tr>
<tr>
<td>reverse</td>
<td>From the most recent entry to the first entry.</td>
</tr>
<tr>
<td>stats</td>
<td>Statistics information.</td>
</tr>
<tr>
<td>tailf</td>
<td>New traces as they are added.</td>
</tr>
<tr>
<td>udir</td>
<td>Copies trace information from remote locations to the specified temporary directory.</td>
</tr>
</tbody>
</table>
The following command displays Type 6 password encryption feature information:

```
Router# show type6 server
Server detail information:
--------------------------
AES config State : Enabled
Masterkey config State : Enabled
Type6 feature State : Enabled
Master key Inprogress : No
```

```
Router# show type6 trace server all
Client file lib/type6/type6_server_wr
25 wrapping entries (18496 possible, 64 allocated, 0 filtered, 25 total)
Jul 19 09:59:27.168 lib/type6/type6_server_wr 0/RP0/CPU0 t7145 ***** Type6 server process
started Respawn count (1) ****
...
Jul 19 12:22:59.908 lib/type6/type6_server_wr 0/RP0/CPU0 t7145 User has started Master key
operation (CREATE)
Jul 19 12:22:59.908 lib/type6/type6_server_wr 0/RP0/CPU0 t7145 Created Master key in TAM
successfully
Jul 19 12:23:00.265 lib/type6/type6_server_wr 0/RP0/CPU0 t7145 Master key Available set to
(AVAILABLE)
Jul 19 12:23:00.272 lib/type6/type6_server_wr 0/RP0/CPU0 t7145 Master key inprogress set
to (NOT INPROGRESS)
```

```
Router# show type6 clients
Type6 Clients information:

Client Name   MK State
---------------
keychain      UNKNOWN
```
show type6
Management Plane Protection Commands

This module describes the commands used to configure management plane protection (MPP).

Note

All commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router that is introduced from Cisco IOS XR Release 6.3.2. References to earlier releases in Command History tables apply to only the Cisco NCS 5500 Series Router.

Note

• Starting with Cisco IOS XR Release 6.6.25, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 560 Series Routers.

• Starting with Cisco IOS XR Release 6.3.2, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router.

• References to releases before Cisco IOS XR Release 6.3.2 apply to only the Cisco NCS 5500 Series Router.

• Cisco IOS XR Software Release 7.0.1 specific updates are not applicable for the following variants of Cisco NCS 540 Series Routers:
  • N540-28Z4C-SYS-A
  • N540-28Z4C-SYS-D
  • N540X-16Z4G8Q2C-A
  • N540X-16Z4G8Q2C-D
  • N540-12Z20G-SYS-A
  • N540-12Z20G-SYS-D
  • N540X-12Z16G-SYS-A
  • N540X-12Z16G-SYS-D
For detailed information about keychain management concepts, configuration tasks, and examples, see the Implementing Management Plane Protection chapter in the *System Security Configuration Guide for Cisco NCS 5500 Series Routers.*

**Note**
Currently, only default VRF is supported. VPNv4, VPNv6 and VPN routing and forwarding (VRF) address families will be supported in a future release.

- address ipv4 (MPP), on page 127
- address ipv6 (MPP), on page 128
- allow, on page 129
- allow local-port, on page 131
- inband, on page 133
- interface (MPP), on page 134
- out-of-band, on page 136
- show mgmt-plane, on page 137
- tpa, on page 139
- vrf (MPP), on page 140
address ipv4 (MPP)

To configure the peer IPv4 or IPv6 address in which management traffic is allowed on the interface, use the `address ipv4` command in interface peer configuration mode. To remove the IP address that was previously configured on this interface, use the `no` form of this command.

```
address  {ipv4 | ipv6}
peer-ip-address
| peer-ip-address/length
no address  {ipv4 | ipv6}
peer-ip-address
| peer-ip-address/length
```

**Syntax Description**

- `peer-ip-address` (Required) Peer IPv4 or IPv6 address in which management traffic is allowed on the interface. This address can effectively be the source address of the management traffic that is coming in on the configured interface.

- `peer-ip-address/length` (Required) Prefix of the peer IP address and IPv4 address or IPv6 format:
  - IPv4—`A.B.C.D/length`
  - IPv6—`X.X:X.X`  

**Command Default**

If no specific peer is configured, all peers are allowed.

**Command Modes**

Interface peer configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>system read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to configure the peer address for management traffic:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# control-plane
RP/0/RP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0/CPU0:router(config-mpp)# inband
RP/0/RP0/CPU0:router(config-mpp-inband)# interface all
RP/0/RP0/CPU0:router(config-mpp-inbandoutband-all)# allow all peer
RP/0/RP0/CPU0:router(config-telnetftppeer)# address ipv4 10.1.0.0/16
```
address ipv6 (MPP)

To configure the peer IPv6 address in which management traffic is allowed on the interface, use the `address ipv6` command in interface peer configuration mode. To remove the IP address that was previously configured on this interface, use the `no` form of this command.

```
address ipv6 {peer-ip-address | peer-ip-address/length}
```

**Syntax Description**

- **peer-ip-address**
  
  Peer IPv6 address in which management traffic is allowed on the interface. This address can effectively be the source address of the management traffic that is coming in on the configured interface.

- **peer-ip-address/length**
  
  Prefix of the peer IPv6 address.

**Command Default**

If no specific peer is configured, all peers are allowed.

**Command Modes**

Interface peer configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

- **system read, write**

**Examples**

The following example shows how to configure the peer IPv6 address 33::33 for management traffic:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# control-plane
RP/0/RP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0/CPU0:router(config-mpp)# inband
RP/0/RP0/CPU0:router(config-mpp-outband)# interface GigabitEthernet 0/1/1/2
RP/0/RP0/CPU0:router(config-mpp-outband-GigabitEthernet0_1_1_2)# allow TFTP peer
RP/0/RP0/CPU0:router(config-tftp-peer)# address ipv6 33::33
```
allow

To configure an interface as an inband or out-of-band interface to allow all peer addresses for a specified protocol or all protocols, use the `allow` command in management plane protection inband interface configuration mode or management plane protection out-of-band interface configuration.

To disallow a protocol on an interface, use the `no` form of this command.

```
allow {protocol | all} [peer]
no allow {protocol | all} [peer]
```

**Syntax Description**

- `protocol` Interface configured to allow peer-filtering for the following specified protocol’s traffic:
  - HTTP(S)
  - NETCONF (version 1.1 protocol)
  - SNMP (also versions)
  - Secure Shell (v1 and v2)
  - TFTP
  - Telnet
  - XML

- `all` Configures the interface to allow peer-filtering for all the management traffic that is specified in the list of protocols.

- `peer` (Optional) Configures the peer address on the interface. Peer refers to the neighboring router interface in which traffic might arrive to the main router.

**Command Default**

By default, no management protocol is allowed on any interface except the management interfaces.

**Command Modes**

Management plane protection inband interface configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

If you permit or allow a specific protocol to an interface, traffic is allowed only for that protocol, and all other management traffic is dropped.

The IOS XR XML API provides a programmatic interface to the router for use by external management applications. This interface provides a mechanism for router configuration and monitoring utilizing XML formatted request and response streams. As one of the management services, XML should be capable of applying MPP. To secure XML MPP data, XML keyword has been added to the command.

**Task ID**

- `read`
- `write`

---

*System Security Command Reference for Cisco NCS 5500 Series, Cisco NCS 540 Series, and Cisco NCS 560 Series Routers*
The following example shows how to configure all management protocols for all inband interfaces:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# control-plane
RP/0/RP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0/CPU0:router(config-mpp)# inband
RP/0/RP0/CPU0:router(config-mpp-inband)# interface all
RP/0/RP0/CPU0:router(config-mpp-inband-all)# allow all
```

The following example shows how to configure MPP support on an XML peer in-band interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# control-plane
RP/0/RP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0/CPU0:router(config-ctrl-mpp)# inband interface all allow xml peer address ipv4 172.10.10.1
```
allow local-port

To configure a local port and third-party application protocols for management plane protection (MPP) on an interface, use the `allow local-port` command in management plane protection TPA mode. To disallow a protocol on an interface, use the `no` form of this command.

```
allow local-port  port-number  protocol  protocol-number  interface  interface-name  local-address  IP local address  remote-address  IP remote address
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>local-port</td>
<td>Specifies local L4 port of an interface.</td>
</tr>
<tr>
<td>protocol</td>
<td>Specifies the L4 protocol to be configured on MPP.</td>
</tr>
</tbody>
</table>

**Protocol number**
Enter the protocol number corresponding to different protocols. You can choose a value from range 1 to 255. Following are some of the protocol numbers dedicated to different protocols:

- `gre` - Generic Routing Encapsulation. (47)
- `udp` - User Datagram Protocol, RFC 768. (17)
- `tcp` - Transmission Control Protocol, RFC 793. (6)
- `pptp` - Point-to-Point Tunneling Protocol. Entering the pptp protocol literal is equivalent to entering the gre protocol literal. (47)
- `pim` - Protocol Independent Multicast. (103)
- `ospf` - Open Shortest Path First routing protocol, RFC 1247. (89)
- `ipsec` - IP Security. Entering the ipsec protocol literal is equivalent to entering the esp protocol literal. (50)
- `ipinip` - IP-in-IP encapsulation. (4)
- `icmp6` - Internet Control Message Protocol for IPv6, RFC 2463. (58)
- `igmp` - Internet Group Management Protocol, RFC 1112. (2)
- `igrp` - Interior Gateway Routing Protocol. (9)

**Note**
In IOS XR release 6.5.2, protocol number is replaced by protocol names. The supported protocols are `tcp` and `udp`.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
<td>Specify the MPP interface on which the protocol has to be configured.</td>
</tr>
<tr>
<td>local-address</td>
<td>Specify the local IP address of the host or client.</td>
</tr>
<tr>
<td>remote-address</td>
<td>Specify the remote IP address of the host or client.</td>
</tr>
</tbody>
</table>

**Command Default**

Not Applicable

**Command Modes**

Management plane protection TPA
allow local-port

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.3.2</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Example

Router(config)# control-plane
Router(config-ctrl)# management-plane
Router(config-mpp)# tpa vrf default address-family [ipv4 | ipv6]
Router(config-mpp-tpa-vrf-afi)# allow local-port 57600 protocol tcp interface mgmtEth 0/RP0/CP0/0 local-address 1.1.1.1/32 remote-address 2.2.2.2/32
inband

To configure an inband interface and to enter management plane protection inband configuration mode, use the `inband` command in management plane protection configuration mode. To disable all configurations under inband configuration mode, use the `no` form of this command.

```
inband
no inband
```

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

Management plane protection inband configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `inband` command to enter management plane protection inband configuration mode.

**Examples**

The following example shows how to enter management plane protection inband configuration mode using the `inband` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# control-plane
RP/0/RP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0/CPU0:router(config-mpp)# inband
RP/0/RP0/CPU0:router(config-mpp-inband)#
```
interface (MPP)

To configure a specific interface or all interfaces as an inband or out-of-band interface, use the `interface` command in management plane protection inband configuration mode or management plane protection out-of-band configuration mode.

To disable all the configurations under an interface mode, use the `no` form of this command.

```
interface {type interface-path-id | all}
no interface {type interface-path-id | all}
```

**Syntax Description**

- `type`: Interface type. For more information, use the question mark (?) online help function.
- `interface-path-id`: Virtual interface instance. Number range varies depending on interface type.

**Note**

Use the `show interfaces` command in EXEC mode to see a list of all interfaces currently configured on the router.

For more information about the syntax for the router, use the question mark (?) online help function.

`all`: Configures all interfaces to allow for management traffic.

**Command Default**

None

**Command Modes**

Management plane protection out-of-band configuration
Management plane protection inband configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `interface` command to enter management plane protection inband interface configuration mode or management plane protection out-of-band interface configuration mode.

For the `instance` argument, you cannot configure Management Ethernet interfaces as inband interfaces.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to configure all inband interfaces for MPP:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# control-plane
RP/0/RP0/CPU0:router(config-ctrl)# management-plane
```
RP/0/RP0/CPU0:router(config-mpp)# inband
RP/0/RP0/CPU0:router(config-mpp-inband)# interface all
RP/0/RP0/CPU0:router(config-mpp-inband-all)#

The following example shows how to configure all out-of-band interfaces for MPP:

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# control-plane
RP/0/RP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0/CPU0:router(config-mpp)# out-of-band
RP/0/RP0/CPU0:router(config-mpp-outband)# interface all
RP/0/RP0/CPU0:router(config-mpp-outband-all)#
To configure out-of-band interfaces or protocols and to enter management plane protection out-of-band configuration mode, use the `out-of-band` command in management plane protection configuration mode. To disable all configurations under management plane protection out-of-band configuration mode, use the `no` form of this command.

### Syntax Description
This command has no keywords or arguments.

### Command Default
None

### Command Modes
Management plane protection out-of-band configuration

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines
Use the `out-of-band` command to enter management plane protection out-of-band configuration mode.

*Out-of-band* refers to an interface that allows only management protocol traffic to be forwarded or processed. An *out-of-band management interface* is defined by the network operator to specifically receive network management traffic. The advantage is that forwarding (or customer) traffic cannot interfere with the management of the router.

### Examples
The following example shows how to enter management plane protection out-of-band configuration mode using the `out-of-band` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# control-plane
RP/0/RP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0/CPU0:router(config-mpp)# out-of-band
RP/0/RP0/CPU0:router(config-mpp-outband)#
```
show mgmt-plane

To display information about the management plane such as type of interface and protocols enabled on the interface, use the `show mgmt-plane` command.

```
show mgmt-plane [ { inband | out-of-band } ] [ { interface type interface-path-id | vrf } ]
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inband</td>
<td>(Optional) Displays the inband management interface configurations that are the interfaces that process management packets as well as data-forwarding packets. An inband management interface is also called a <em>shared management interface</em>.</td>
</tr>
<tr>
<td>out-of-band</td>
<td>(Optional) Displays the out-of-band interface configurations. Out-of-band interfaces are defined by the network operator to specifically receive network management traffic.</td>
</tr>
<tr>
<td>interface</td>
<td>(Optional) Displays all the protocols that are allowed in the specified interface.</td>
</tr>
<tr>
<td>type</td>
<td>Interface type. For more information, use the question mark (?) online help function.</td>
</tr>
<tr>
<td>interface-path-id</td>
<td>Interface instance. Number range varies depending on interface type.</td>
</tr>
<tr>
<td>vrf</td>
<td>(Optional) Displays the Virtual Private Network (VPN) routing and forwarding reference of an out-of-band interface.</td>
</tr>
</tbody>
</table>

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

| Release 6.0 | This command was introduced. |

**Usage Guidelines**

The `vrf` keyword is valid only for out-of-band VRF configurations.

**Task ID**

```
Task ID Operations
system read
```

**Examples**

The following sample output displays all the interfaces that are configured as inband or out-of-band interfaces under MPP:

```
RP/0/RP0/CPU0:router# show mgmt-plane
```
Management Plane Protection

inband interfaces
-------------------
interface - HundredGigabitEthernet0_1_1_0
  ssh configured -
    All peers allowed
telnet configured -
  peer v4 allowed - 10.1.0.0/16
  all configured -
    All peers allowed
interface - HundredGigabitEthernet0_1_1_0
telnet configured -
  peer v4 allowed - 10.1.0.0/16

interface - all
  all configured -
    All peers allowed

outband interfaces
------------------
interface - HundredGigabitEthernet0_1_1_0
tftp configured -
  peer v6 allowed - 33::33

The following sample output displays the Virtual Private Network (VPN) routing and forwarding (VRF) reference of an out-of-band interface:

RP/0/RP0/CPU0:router# show mgmt-plane out-of-band vrf

Management Plane Protection -
  out-of-band VRF - my_out_of_band
tpa

To configure a third-party application protocol for Management Plane Protection (MPP), use the tpa command in management plane protection configuration mode. To disable all configurations related to the third-party application, use the no form of this command.

```
tpa vrf default address-family [ipv4 | ipv6]
```

### Syntax Description

- **vrf**: Configures a Virtual Private Network (VPN) routing and forwarding (VRF) reference.
- **address-family**: Enables support for various address family configuration modes while configuring TPA.
- **ipv4**: Specifies IP Version 4 address prefixes.
- **ipv6**: Specifies IP Version 6 address prefixes.

### Command Default
Not Applicable

### Command Modes
Management plane protection configuration

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.3.2</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines
Only default vrf is supported for TPA configuration.

### Example

```
Router(config)# control-plane
Router(config-ctrl)# management-plane
Router(config-mpp)# tpa vrf default address-family [ipv4 | ipv6]
```
vrf (MPP)

To configure a Virtual Private Network (VPN) routing and forwarding (VRF) reference of an out-of-band interface, use the `vrf` command in management plane protection out-of-band configuration mode. To remove the VRF definition before the VRF name is used, use the `no` form of this command.

```
vrf vrf-name
no vrf vrf-name
```

### Syntax Description

- **vrf-name**: Name assigned to a VRF.

### Command Default

The VRF concept must be used to configure interfaces as out-of-band. If no VRF is configured during an out-of-band configuration, the interface goes into a default VRF.

### Command Modes

Management plane protection out-of-band configuration

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

If the VRF reference is not configured, the default name MPP_OUTBAND_VRF is used. If there is an out-of-band configuration that is referring to a VRF and the VRF is deleted, all the MPP bindings are removed.

### Examples

The following example shows how to configure the VRF:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# vrf my_out_of_band
RP/0/RP0/CPU0:router(config-vrf)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-vrf-af)# exit
RP/0/RP0/CPU0:router(config-vrf)# address-family ipv6 unicast
RP/0/RP0/CPU0:router(config-vrf-af)# commit
RP/0/RP0/CPU0:router(config-vrf-af)# end
RP/0/RP0/CPU0:router#
```

The following example shows how to configure the VRF definition for MPP:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# control-plane
RP/0/RP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0/CPU0:router(config-mpp)# out-of-band
RP/0/RP0/CPU0:router(config-mpp-outband)# vrf my_out_of_band
```
Traffic Protection Commands

This module describes the commands used to configure traffic protection.

---

Note

All commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router that is introduced from Cisco IOS XR Release 6.3.2. References to earlier releases in Command History tables apply to only the Cisco NCS 5500 Series Router.

---

Note

- Starting with Cisco IOS XR Release 6.6.25, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 560 Series Routers.
- Starting with Cisco IOS XR Release 6.3.2, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router.
- References to releases before Cisco IOS XR Release 6.3.2 apply to only the Cisco NCS 5500 Series Router.
- Cisco IOS XR Software Release 7.0.1 specific updates are not applicable for the following variants of Cisco NCS 540 Series Routers:
  - N540-28Z4C-SYS-A
  - N540-28Z4C-SYS-D
  - N540X-16Z4G8Q2C-A
  - N540X-16Z4G8Q2C-D
  - N540-12Z20G-SYS-A
  - N540-12Z20G-SYS-D
  - N540X-12Z16G-SYS-A
  - N540X-12Z16G-SYS-D
For detailed information about traffic protection concepts, configuration tasks, and examples, see the *Traffic Protection for Third-Party Applications* chapter in the *System Security Configuration Guide for Cisco NCS 5500 Series Routers*.

- allow, on page 143
- tpa, on page 145
allow

To configure a local port and third-party application protocols for traffic protection, use the allow command in protection mode. To disallow a protocol on an interface, use the no form of this command.

```
allow protocol { tcp | udp } local-port port-number { interface interface-name | local-address local IP address | remote-address remote IP address }
```

```
no allow protocol { tcp | udp } local-port port-number { interface interface-name | local-address local IP address | remote-address remote IP address }
```

### Syntax Description

- **protocol**: Specifies the L4 protocol to be configured for traffic protection. The supported protocols are TCP and UDP.
- **local-port**: Specifies local L4 port.
- **Port-number**: Specifies a port number in the range of 1 to 65535.
- **interface**: Specifies the interface on which the protocol has to be configured.
- **local-address**: Specifies the local IP address of the host or client.
- **remote-address**: Specifies the remote IP address of the host or client.

### Command Default

Not Applicable

### Command Modes

Protection

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5.2</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

If no allow command is used for a given local port and protocol, then by default, any ingress traffic is delivered to Third Party Applications. If one or more allow entries are added, only the ingress traffic matching an allow entry is delivered for that protocol and port. It is possible to configure multiple allow entries for the same protocol and port, for example, to allow traffic from multiple remote addresses.

#### Note

If multiple allow entries are configured for the same protocol and port, the entries are expected to be non-overlapping. If overlapping entries are present, for example, multiple remote addresses in overlapping subnets, then the behaviour is platform-dependent.
Example

The following example shows how to configure a local port and third-party application protocols for traffic protection:

Router# configure
Router(config)# tpa
Router(config-tpa)# vrf default
Router(config-tpa-vrf)# address-family ipv4
Router(config-tpa-vrf-afi)# protection
Router(config-tpa-vrf-afi-prot)# allow protocol tcp local-port 6 remote-address 192.0.2.3
interface MgmtEth0 local-address 192.0.2.125
To configure a third-party application protocol for traffic protection, use the `tpa` command in global configuration mode. To disable all configurations related to the third-party application, use the `no` form of this command.

```
tpa vrf vrf-name address-family [ipv4 | ipv6] protection
no tpa vrf vrf-name address-family [ipv4 | ipv6] protection
```

### Syntax Description
- **vrf**
  - Configures a VPN routing and forwarding (VRF) reference.
- **address-family**
  - Enables support for various address family configuration modes while configuring TPA.
- **ipv4**
  - Specifies IP Version 4 address prefixes.
- **ipv6**
  - Specifies IP Version 6 address prefixes.
- **protection**
  - Enters the Traffic Protection submode.

### Command Default
Not Applicable

### Command Modes
Global configuration

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines
Some platforms do not support non-management traffic in any VRFs apart from default VRF.

### Example
The following example shows how to configure a third-party application protocol for traffic protection.

```
Router# configure
Router(config)# tpa
Router(config-tpa)# vrf vrf-name
Router(config-tpa-vrf)# address-family [ipv4 | ipv6]
Router(config-tpa-vrf-afi)# protection
```
tpa
802.1X Authentication Commands

This module describes the commands used for 802.1X Authentication.

Note

All commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router that is introduced from Cisco IOS XR Release 6.3.2. References to earlier releases in Command History tables apply to only the Cisco NCS 5500 Series Router.

Note

• Starting with Cisco IOS XR Release 6.6.25, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 560 Series Routers.

• Starting with Cisco IOS XR Release 6.3.2, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router.

• References to releases before Cisco IOS XR Release 6.3.2 apply to only the Cisco NCS 5500 Series Router.

• Cisco IOS XR Software Release 7.0.1 specific updates are not applicable for the following variants of Cisco NCS 540 Series Routers:
  • N540-28Z4C-SYS-A
  • N540-28Z4C-SYS-D
  • N540X-16Z4G8Q2C-A
  • N540X-16Z4G8Q2C-D
  • N540-12Z20G-SYS-A
  • N540-12Z20G-SYS-D
  • N540X-12Z16G-SYS-A
  • N540X-12Z16G-SYS-D

This module provides command line interface (CLI) commands for 802.1X Authentication Commands.
For detailed information about 802.1X authentication commands, configuration tasks, and examples, see the 802.1X Port-Based Authentication chapter in the System Security Configuration Guide for Cisco NCS 5500 Series Routers.

- dot1x host-mode, on page 149
- show dot1x, on page 150
dot1x host-mode

To allow multiple hosts or MAC addresses on a single port, use the host-mode command under authenticator mode in dot1x profile.

```
host-mode { multi-auth | multi-host | single-host }
```

**Syntax Description**

- **multi-auth** Multiple authentication mode
- **multi-host** Multiple host mode
- **single-host** Single host mode

**Command Default**
The default is **multi-auth mode**.

**Command Modes**
XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.2.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Use the following steps to configure 802.1X host-modes:

```
Router# configure terminal
Router(config)# dot1x profile {name}
Router(config-dot1x-auth)# pae {authenticator}
Router(config-dot1x-auth-auth)# host-mode
multi-auth multiple authentication mode
multi-host multiple host mode
single-host single host mode
```
show dot1x

To display whether 802.1X authentication has been configured on the device, use the `show dot1x` command in privileged EXEC mode.

```
show dot1x [interface interface-type interface-id | detail]
```

**Syntax Description**

- `interface interface-type interface-id` Displays the information for the specified interface ID.

**Command Default**

None

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.6.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation Task ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>dot1x</td>
<td>read</td>
</tr>
</tbody>
</table>

**Example**

The `show dot1x interface` command verifies whether the 802.1X port-based authentication is successful or not for the supplicant to proceed with the traffic flow on the configured interface.

```
Router# show dot1x interface HundredGigE 0/0/1/0 detail
```

```
Dot1x info for HundredGigE 0/0/1/0
================================================================================
Interface short name : Hu0/0/1/0
Interface handle : 0x4080
Interface MAC : 021a.9eeb.6a59
Ethertype : 888E
PAE : Authenticator
**Dot1x Port Status** : AUTHORIZED
Dot1x Profile : test_prof
L2 Transport : FALSE
Authenticator:
  Port Control : Enabled
  Config Dependency : Resolved
  Eap profile : None
  ReAuth : Disabled
Client List:
  Supplicant : 027E.15F2.CAE7
**Programming Status** : Add Success
**Auth SM State** : Authenticated
Auth Bend SM State : Idle
Last authen time : 2018 Dec 11 17:00:30.912
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last authen server</td>
<td>Remote radius server</td>
</tr>
<tr>
<td>Time to next reauth</td>
<td>reauth not enabled</td>
</tr>
<tr>
<td>MKA Interface:</td>
<td></td>
</tr>
<tr>
<td>Dot1x Tie Break Role</td>
<td>NA (Only applicable for PAE role both)</td>
</tr>
<tr>
<td>EAP Based Macsec</td>
<td>Disabled</td>
</tr>
<tr>
<td>MKA Start time</td>
<td>NA</td>
</tr>
<tr>
<td>MKA Stop time</td>
<td>NA</td>
</tr>
<tr>
<td>MKA Response time</td>
<td>NA</td>
</tr>
</tbody>
</table>
show dot1x
MACsec Commands

This module describes the commands used to configure MACsec.

Note

All commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router that is introduced from Cisco IOS XR Release 6.3.2. References to earlier releases in Command History tables apply to only the Cisco NCS 5500 Series Router.

For detailed information about keychain management concepts, configuration tasks, and examples, see the Implementing MACsec encryption chapter in the System Security Configuration Guide for Cisco NCS 5500 Series Routers.

- cipher-suite, on page 154
- conf-offset, on page 155
- cryptographic-algorithm, on page 156
- fallback-psk-keychain, on page 158
- key, on page 159
- key chain, on page 160
- key-string, on page 161
- key-server-priority, on page 163
- lifetime, on page 164
- macsec, on page 166
- macsec-policy, on page 168
- macsec shutdown, on page 169
- security-policy, on page 170
- window-size, on page 171
cipher-suite

Configures the cipher suite for encrypting traffic with MACsec in the MACsec policy configuration mode.

The first portion of the cipher name indicates the encryption method, the second portion indicates the hash or integrity algorithm, and the third portion indicates the length of the cipher (128/256).

To disable this feature, use the no form of this command.

cipher-suite encryption_suite

**Syntax Description**

- `encryption_suite` The GCM encryption method that uses the AES encryption algorithm. The available encryption suites are:
  - GCM-AES-128
  - GCM-AES-256
  - GCM-AES-XPN-128
  - GCM-AES-XPN-256

**Command Default**

The default cipher suite chosen for encryption is GCM-AES-XPN-256.

**Command Modes**

MACsec policy configuration.

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to use the `cipher-suite` command:

```
RP/0/RP0/CPU0:router# configure t
RP/0/RP0/CPU0:router(config)# macsec-policy mac_policy
RP/0/RP0/CPU0:router(config-mac_policy)# cipher-suite GCM-AES-XPN-256
RP/0/RP0/CPU0:router(config-mac_policy)#
```
conf-offset

Configures the confidentiality offset for MACsec encryption in the MACsec policy configuration mode.
To disable this feature, use the no form of this command.

```
conf-offset offset_value
```

**Syntax Description**

- `offset_value` Configures the offset value. The options are:
  - CONF-OFFSET-0: Does not offset the encryption
  - CONF-OFFSET-30: Offsets the encryption by 30 characters
  - CONF-OFFSET-50: Offsets the encryption by 50 characters.

**Command Default**

Default value is 0.

**Command Modes**

MACsec policy configuration.

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**

- `system` read, write

**Examples**

The following example shows how to use the `conf-offset` command:

```
RP/0/RP0/CPU0:router# configure t
RP/0/RP0/CPU0:router(config)# macsec-policy mac_policy
RP/0/RP0/CPU0:router(config-mac_policy)# conf-offset CONF-OFFSET-30
RP/0/RP0/CPU0:router(config-mac_policy)#
```
cryptographic-algorithm

Configures the cryptographic algorithm used for authenticating a peer for MACsec encryption in the Keychain-key configuration mode.

To disable this feature, use the no form of this command.

cryptographic-algorithm authentication algorithm

Syntax Description

authentication algorithm Configures the 128-bit or 256-bit AES encryption algorithm.

Command Default

No default behavior or values.

Command Modes

Keychain-key configuration.

Usage Guidelines

If you do not specify the cryptographic algorithm, MAC computation and API verification would be invalid.

Examples

The following example shows how to use the cryptographic-algorithm command for MACsec Encryption:

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router# key chain mac_chain macsec
RP/0/RP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string
11111111111111111111111111111111 cryptographic-algorithm aes-256-cmac

Examples

The following example shows how to use the AES-128-CMAC authentication algorithm command:

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router# key chain mac_chain macsec
RP/0/RP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string
12345678123456781234567812345678 cryptographic-algorithm aes-128-cmac

Examples

The following example shows how to use the AES-256-CMAC authentication algorithm command:

RP/0/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router# key chain mac_chain macsec
RP/0/RP0/CPU0:router(config-mac_chain-MacSec) # key 1234abcd5678
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string
123456781234567812345678123456781234567812345678123456781234567812345678
cryptographic-algorithm
aes-256-cmac
**fallback-psk-keychain**

To create or modify a fallback psk keychain key, use the `fallback-psk-keychain` command in keychain-key configuration mode.

To disable this feature, use the `no` form of this command.

```
fallback-psk-keychain  key-id
```

**Syntax Description**

- `key-id` 64-character hexadecimal string.

**Command Default**

No default behavior or values.

**Command Modes**

Key chain configuration

**Usage Guidelines**

The key must be of even number of characters. Entering an odd number of characters will exit the MACsec configuration mode.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to use the `key` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router# fallback-psk-keychain fallback_mac_chain
RP/0/RP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
```
key

To create or modify a keychain key, use the **key** command in keychain-key configuration mode. To disable this feature, use the **no** form of this command.

```
key key-id
```

**Syntax Description**
- `key-id` 64-character hexadecimal string.

**Command Default**
No default behavior or values.

**Command Modes**
Key chain configuration

**Usage Guidelines**
The key must be of even number of characters. Entering an odd number of characters will exit the MACsec configuration mode.

**Task ID**
<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**
The following example shows how to use the **key** command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router# key chain mac_chain macsec
RP/0/RP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
```
**key chain**

To create or modify a keychain, use the **key chain** command in the key chain configuration mode.

To disable this feature, use the **no** form of this command.

```plaintext
key chain key-chain-name
```

**Syntax Description**

`key-chain-name` Specifies the name of the keychain. The maximum length is 32 (128-bit encryption)/64 (256-bit encryption) character hexadecimal string.

**Note** If you are configuring MACsec to interoperate with a MACsec server that is running software prior to IOS XR 6.1.3, then ensure that the MACsec key length is of 64 characters. If the key length is lesser than 64 characters, authentication will fail.

**Command Modes**

Key chain configuration

**Command Default**

No default behavior or values

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how you can configure a key chain for MACsec encryption:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0/CPU0:router(config-mac_chain-MacSec)#
```
key-string

To specify the text string for the key, use the key-string command in keychain-key configuration mode. To disable this feature, use the no form of this command.

```
key-string [clear | password] key-string-text
```

**Syntax Description**

- **clear**: Specifies the key string in clear-text form.
- **password**: Specifies the key in encrypted form.
- **key-string-text**: Text string for the key, which is encrypted by the parser process before being saved to the configuration. The text string has the following character limitations:
  - Plain-text key strings—Minimum of 1 character and a maximum of 32 (128-bit encryption)/64 (256-bit encryption) characters (hexadecimal string).
  - Encrypted key strings—Minimum of 4 characters and no maximum.

**Command Default**

The default value is clear.

**Command Modes**

Key chain configuration

**Usage Guidelines**

For an encrypted password to be valid, the following statements must be true:

- String must contain an even number of characters, with a minimum of four.
- The first two characters in the password string must be decimal numbers and the rest must be hexadecimals.
- The first two digits must not be a number greater than 53.

Either of the following examples would be valid encrypted passwords:

```
1234abcd
```

or

```
50aefd
```

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to use the keystring command:

```
! For AES 128-bit encryption
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
```
MACsec Commands
key-string

RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string
12345678123456781234567812345678 cryptographic-algorithm AES-128-CMAC
! For AES 256-bit encryption
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string
1234567812345678123456781234567812345678123456781234567812345678 cryptographic-algorithm
AES-256-CMAC

System Security Command Reference for Cisco NCS 5500 Series, Cisco NCS 540 Series, and Cisco NCS 560 Series Routers
162


key-server-priority

Configures the preference for a device to serve as the key server for MACsec encryption in the MACsec policy configuration mode. To disable this feature, use the no form of this command.

**key-server-priority**  *value*

**Syntax Description**

*value*  Indicates the priority for a device to become the key server. Lower the value, higher the preference.

The range is 0-255.

**Command Default**

Default value is 16.

**Command Modes**

MACsec policy configuration.

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to use the **key-server-priority** command:

```
RP/0/RP0/CPU0:router# configure t
RP/0/RP0/CPU0:router(config)# macsec-policy mac_policy
RP/0/RP0/CPU0:router(config-mac_policy)# key-server-priority 16
RP/0/RP0/CPU0:router(config-mac_policy)#
```
**lifetime**

Configures the validity period for the MACsec key or CKN in the Keychain-key configuration mode. To disable this feature, use the no form of this command.

The lifetime period can be configured with a duration in seconds, as a validity period between two dates (for example, Jan 01 2014 to Dec 31 2014), or with an infinite validity.

The key is valid from the time you configure in HH:MM:SS format. Duration is configured in seconds.

When a key has expired, the MACsec session is torn down and running the `show macsec mka session` command does not display any information. If you run the `show macsec mka interface` and `show macsec mka interface detail` commands, you can see that the session is unsecured.

```
lifetime start_time start_date
    { end_time end_date | duration validity | infinite }
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start-time</td>
<td>Start time in hh:mm:ss from which the key becomes valid. The range is from 0:0:0 to 23:59:59.</td>
</tr>
<tr>
<td>end-time</td>
<td>End time in hh:mm:ss at which point the key becomes invalid. The range is from 0:0:0 to 23:59:59.</td>
</tr>
<tr>
<td>start_date</td>
<td>The date in DD month YYYY format that the key becomes valid.</td>
</tr>
<tr>
<td>end_date</td>
<td>The date in DD month YYYY format that the key becomes invalid.</td>
</tr>
<tr>
<td>duration</td>
<td>The key chain is valid for the duration you configure. You can configure duration in seconds.</td>
</tr>
<tr>
<td>validity</td>
<td>The key chain is valid indefinitely.</td>
</tr>
</tbody>
</table>

**Command Default**

No default behavior or values

**Command Modes**

Keychain-key configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>
The following example shows how to use the `lifetime` command:

**! For AES 128-bit encryption**

```bash
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string
12345678123456781234567812345678123456781234567812345678123456781234567812345678
cryptographic-algorithm AES-128-CMAC
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# lifetime 05:00:00 20 february 2015 12:00:00 30 september 2016
```

**! For AES 256-bit encryption**

```bash
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string
1234567812345678123456781234567812345678123456781234567812345678123456781234567812345678123456781234567812345678123456781234567812345678123456781234567812345678123456781234567812345678123456781234567812345678123456781234567812345678123456781234567812345678
cryptographic-algorithm AES-256-CMAC
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# lifetime 05:00:00 20 february 2015 12:00:00 30 september 2016
```
macsec

Enables MACsec on the router in the keychain configuration mode. To disable this feature, use the no form of this command.

macsec [key key-id ]

Syntax Description

key-id The key can be up to 64 bytes in length. The configured key is the CKN that is exchanged between the peers.

Command Default

No default behavior or values.

Command Modes

Keychain configuration

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 5.3.2</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>Release 7.1.2</td>
<td>The key-id values are made case insensitive, and are stored as uppercase letters.</td>
</tr>
<tr>
<td>Release 7.2.1</td>
<td></td>
</tr>
</tbody>
</table>

Usage Guidelines

From Cisco IOS XR Software Release 7.1.2, Release 7.2.1 and later, the MACsec key IDs are considered to be case insensitive. These key IDs are stored as uppercase letters. For example, a key ID of value 'FF' and of value 'ff' are considered to be the same, and both these key IDs are now stored in uppercase as 'FF'. Whereas, prior to Release 7.1.2 and Release 7.2.1, both these values were treated as case sensitive, and hence considered as two separate key IDs. However, the support for this case insensitive IDs is applicable only for the configurations done through CLI, and not for configurations done through Netconf protocol. Hence, it is recommended to have unique strings as key IDs for a MACsec key chain to avoid flapping of MACsec sessions.

For example, the key IDs ('FF' and 'ff') in this example are not unique (although one is in uppercase and other is in lowercase), and hence this might cause a MACsec session flap.

```plaintext
key chain 1
macsec
key FF
  lifetime 02:01:01 may 18 2020 infinite
! key ff
  lifetime 01:01:01 may 18 2020 infinite
```

Examples

The following example shows how to use the macsec command:
RP/0/RP0/CPU0:router# configure t
RP/0/RP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#
macsec-policy

Creates a MACsec policy for MACsec encryption in XR Config mode. To disable this feature, use the no form of this command.

`macsec-policy  policy_name`

**Syntax Description**
- `policy_name` Name of the MACsec policy for encryption.

**Command Default**
No default behavior or values.

**Command Modes**
XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**
- **Task** system
- **Operations** read, write

**Examples**
The following example shows how to use the `macsec-policy` command:

```
RP/0/RP0/CPU0:router# configure t
RP/0/RP0/CPU0:router(config)# macsec-policy mac_policy
RP/0/RP0/CPU0:router(config-mac_policy)#
```
macsec shutdown

To enable MACsec shutdown, use the **macsec shutdown** command in XR Config mode. To disable MACsec shutdown, use the **no** form of the command.

**macsec shutdown**

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

The **macsec shutdown** command is disabled by default.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3.3</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Enabling the **macsec shutdown** command, brings down all macsec sessions on the MACsec-enabled interfaces and resets ports to non-macsec mode. The already existing MACsec configurations remain unaffected by enabling this feature.

Disabling the **macsec shutdown** command, brings up MACsec sessions for the configured interfaces and enforces MACsec policy on the port.

**Warning**

Configuring **macsec shutdown** command disables MACsec on all data ports, system wide. Execute **clear** command to erase cached configuration or **commit** command to continue.

**Task ID**

- **Task ID**
  - system  read,
  - system  write

**Example**

The following example shows how to enable MACsec shutdown:

```
RP/0/RSP0/CPU0:router# configure terminal
RP/0/RSP0/CPU0:router(config)# macsec shutdown
```
security-policy

Configures the type of data that is allowed to transit out of the interface configured with MACsec in the MACsec policy configuration mode. To disable this feature, use the no form of this command.

security-policy {should-secure | must-secure}

Syntax Description

should-secure  Configures the interface on which the MACsec policy is applied, to permit all data.

must-secure   Configures the interface on which the MACsec policy is applied, to permit only MACsec encrypted data.

Command Default

Default value is must-secure.

Command Modes

MACsec policy configuration.

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

Examples

The following example shows how to use the security-policy command:

```
RP/0/RP0/CPU0:router# configure t
RP/0/RP0/CPU0:router(config)# macsec-policy mac_policy
RP/0/RP0/CPU0:router(config-mac_policy)# security-policy must-secure
RP/0/RP0/CPU0:router(config-mac_policy)#
```
window-size

Configures the replay protection window size in MACsec policy configuration mode. To disable this feature, use the no form of this command.

The replay protection window size indicates the number of out-of-sequence frames that can be accepted at the interface configured with MACsec, without being dropped.

**window-size value**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>value Number of out-of-sequence frames that can be accepted at the interface without being dropped. The range is 0-1024.</th>
</tr>
</thead>
</table>

**Command Default**

Default value is 64.

**Command Modes**

MACsec policy configuration.

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to use the `window-size` command:

```
RP/0/RP0/CPU0:router# configure t
RP/0/RP0/CPU0:router(config)# macsec-policy mac_policy
RP/0/RP0/CPU0:router(config-mac_policy)# window-size 64
```
MACsec Commands

window-size
Public Key Infrastructure Commands

This module describes the commands used to configure Public Key Infrastructure (PKI).

Note

All commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router that is introduced from Cisco IOS XR Release 6.3.2. References to earlier releases in Command History tables apply to only the Cisco NCS 5500 Series Router.

Note

- Starting with Cisco IOS XR Release 6.6.25, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 560 Series Routers.

- Starting with Cisco IOS XR Release 6.3.2, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router.

- References to releases before Cisco IOS XR Release 6.3.2 apply to only the Cisco NCS 5500 Series Router.

- Cisco IOS XR Software Release 7.0.1 specific updates are not applicable for the following variants of Cisco NCS 540 Series Routers:
  - N540-28Z4C-SYS-A
  - N540-28Z4C-SYS-D
  - N540X-16Z4G8Q2C-A
  - N540X-16Z4G8Q2C-D
  - N540-12Z20G-SYS-A
  - N540-12Z20G-SYS-D
  - N540X-12Z16G-SYS-A
  - N540X-12Z16G-SYS-D
For detailed information about PKI concepts, configuration tasks, and examples, see the Implementing Certification Authority Interoperability chapter in the *System Security Configuration Guide for Cisco NCS 5500 Series Routers.*

**Note**
Currently, only default VRF is supported. VPNv4, VPNv6 and VPN routing and forwarding (VRF) address families will be supported in a future release.

- ca-keypair, on page 176
- clear crypto ca certificates, on page 177
- clear crypto ca crl, on page 178
- crl optional (trustpoint), on page 179
- crypto ca authenticate, on page 180
- crypto ca cancel-enroll, on page 182
- crypto ca enroll, on page 183
- crypto ca import, on page 185
- crypto ca trustpoint, on page 186
- crypto ca trustpool import url, on page 188
- crypto ca trustpool policy, on page 189
- crypto key generate dsa, on page 190
- crypto key generate ecdsa, on page 192
- crypto key generate rsa, on page 194
- crypto key import authentication rsa, on page 196
- crypto key zeroize dsa, on page 197
- crypto key zeroize rsa, on page 198
- description (trustpoint), on page 199
- enrollment retry count, on page 200
- enrollment retry period, on page 201
- enrollment terminal, on page 202
- enrollment url, on page 203
- ip-address (trustpoint), on page 205
- key-usage, on page 206
- keypair, on page 208
- lifetime (trustpoint), on page 209
- message-digest, on page 210
- query url, on page 211
- rsakeypair, on page 212
- serial-number (trustpoint), on page 213
- sftp-password (trustpoint), on page 214
- sftp-username (trustpoint), on page 215
- subject-name (trustpoint), on page 216
- show crypto ca certificates, on page 218
- show crypto ca crls, on page 220
- show crypto ca trustpool policy, on page 221
- show crypto key mypubkey dsa, on page 222
- show crypto key mypubkey rsa, on page 223
• show platform security integrity dossier, on page 224
• utility sign, on page 226
To create the key pair for the root certificate on the router, use the `ca-keypair` command in trustpoint configuration mode. To remove this configuration, use the no form of this command.

```
ca-keypair {dsa | ecdsanistp256 | ecdsanistp384 | ecdsanistp521 | rsa} key-pair-label
```

**Syntax Description**

- **key-pair-label**: Specifies the key pair label for the respective key signature algorithm (DSA, ECDSA or RSA).

**Command Default**

None

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Examples**

This example shows how to create the key pair for the root certificate on the router:

```
Router# configure
Router(config)# crypto ca trustpoint system-trustpoint
Router(config-trustp)# ca-keypair rsa system-root-key
Router# commit
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>keypair, on page 208</td>
<td>Creates the key pair for the leaf certificate on the router.</td>
</tr>
</tbody>
</table>
clear crypto ca certificates

To clear certificates associated with trustpoints that no longer exist in the configuration file, use the `clear crypto ca certificates` command in XR EXEC mode.

### Syntax Description

```
clear crypto ca certificates trustpoint
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trustpoint</td>
<td>Trustpoint name.</td>
</tr>
</tbody>
</table>

### Command Default

None

### Command Modes

XR EXEC mode

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

If the router is loaded with a new configuration file and certificates in the new configuration file do not have their corresponding trustpoint configuration, use the `clear crypto ca certificates` command to clear the certificates associated with trustpoints that no longer exist in the configuration file.

The `clear crypto ca certificates` command deletes both certification authority (CA) and router certificates from the system.

### Examples

The following example shows how to clear the certificates associated with trustpoints that no longer exist in the configuration file:

```
RP/0/RP0/CPU0:router# clear crypto ca certificates tp_1
```
clear crypto ca crl

To clear all the Certificate Revocation Lists (CRLs) stored on the router, use the `clear crypto ca crl` command in XR EXEC mode.

**clear crypto ca crl**

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

No default behavior or values

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `clear crypto ca crl` command to clear all CRLs stored on the router. As a result, the router goes through the certification authorities (CAs) to download new CRLs for incoming certificate validation requests.

**Examples**

The following example shows how to clear all CRLs stored on the router:

```
RP/0/RP0/CPU0:router# show crypto ca crls
CRL Entry
Issuer : cn=Certificate Manager,ou=HFR,o=Cisco Systems,l=San Jose,st=CA,c=US
Last Update : [UTC] Wed Jun 5 02:40:04 2002
Next Update : [UTC] Wed Jun 5 03:00:04 2002
CRL Distribution Point : ldap://manager.cisco.com/CN=Certificate Manager,O=Cisco Systems

RP/0/RP0/CPU0:router# clear crypto ca crl
RP/0/RP0/CPU0:router# show crypto ca crls
RP/0/RP0/CPU0:router#
```
crl optional (trustpoint)

To allow the certificates of other peers to be accepted without trying to obtain the appropriate CRL, use the crl optional command in trustpoint configuration mode. To return to the default behavior in which CRL checking is mandatory before your router can accept a certificate, use the no form of this command.

```
crl optional
no crl optional
```

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

The router must have and check the appropriate CRL before accepting the certificate of another IP security peer.

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

When your router receives a certificate from a peer, it searches its memory for the appropriate CRL. If the router finds the appropriate CRL, that CRL is used. Otherwise, the router downloads the CRL from either the certificate authority (CA) or from a CRL distribution point (CDP) as designated in the certificate of the peer. Your router will then check the CRL to ensure that the certificate that the peer sent has not been revoked. If the certificate appears on the CRL, your router cannot accept the certificate and will not authenticate the peer. To instruct the router not to download the CRL and treat the certificate as not revoked, use the crl optional command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example declares a CA and permits your router to accept certificates without trying to obtain a CRL. This example also specifies a nonstandard retry period and retry count.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0/CPU0:router(config-trustp)# enrollment url http://ca_server
RP/0/RP0/CPU0:router(config-trustp)# enrollment retry period 20
RP/0/RP0/CPU0:router(config-trustp)# enrollment retry count 100
RP/0/RP0/CPU0:router(config-trustp)# crl optional
```
crypto ca authenticate

To authenticate the certification authority (CA) by getting the certificate for the CA, use the `crypto ca authenticate` command in XR EXEC mode.

```
crypto ca authenticate {ca-name | system-trustpoint}
```

**Syntax Description**

- `ca-name` Name of the CA Server.
- `system-trustpoint` Generates self-signed root certificate.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td><code>system-trustpoint</code></td>
<td>Generates self-signed root certificate.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `crypto ca authenticate` command is required when you initially configure CA support at your router.

This command authenticates the CA to your router by obtaining the CA certificate, which contains the public key for the CA. For self-signed root CA, because the CA signs its own certificate, you should manually authenticate the CA public key by contacting the CA administrator when you use this command. The certificate fingerprint matching is done out-of-band (for example, phone call, and so forth).

Authenticating a second-level CA requires prior authentication of the root CA.

After the `crypto ca authenticate` command is issued and the CA does not respond by the specified timeout period, you must obtain terminal control again to re-enter the command.

**Examples**

The CA sends the certificate, and the router prompts the administrator to verify the certificate by checking the certificate fingerprint (a unique identifier). The CA administrator can also display the CA certificate fingerprint, so you should compare what the CA administrator sees to what the router displays on the screen. If the fingerprint on the display matches the fingerprint displayed by the CA administrator, you should accept the certificate as valid.

The following example shows that the router requests the CA certificate:

```
Router# crypto ca authenticate msiox
Retrieve Certificate from SFTP server? [yes/no]: yes
Read 860 bytes as CA certificate
```
Subject:
    Name: CA2
    CN= CA2
Issued By :
    cn=CA2
Validity Start : 07:51:51 UTC Wed Jul 06 2005
Validity End : 08:00:43 UTC Tue Jul 06 2010
CRL Distribution Point
    http://10.56.8.236/CertEnroll/CA2.crl
Certificate has the following attributes:
    Fingerprint: D0 44 36 48 CE 08 9D 29 04 C4 2D 69 80 55 53 A3

Do you accept this certificate? [yes/no]: yes

Router#:Apr 10 00:28:52.324 : cepki[335]: %SECURITY-CEPKI-6-INFO : certificate database updated
Do you accept this certificate? [yes/no] yes

This example shows how to generate a self-signed root certificate:

Router#crypto ca authenticate system-trustpoint
crypto ca cancel-enroll

To cancel a current enrollment request, use the `crypto ca cancel-enroll` command in XR EXEC mode.

```
crypto ca cancel-enroll ca-name
```

**Syntax Description**

- `ca-name` Name of the certification authority (CA).

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `crypto ca enroll` command to request certificates from the CA for the Rivest, Shamir, and Adelman (RSA) key pairs for the router defined by the `rsakeypair`, on page 212 command in trustpoint configuration mode. If no `rsakeypair`, on page 212 command is configured for the current trustpoint, the default RSA key pair is used for enrollment. This task is also known as enrolling with the CA. Use the `crypto ca cancel-enroll` command to cancel a current enrollment request.

**Examples**

The following example shows how to cancel a current enrollment request from a CA named `myca`:

```
RP/0/RP0/CPU0:router# crypto ca cancel-enroll myca
```
crypto ca enroll

To obtain a router certificate from the certification authority (CA), use the `crypto ca enroll` command in XR EXEC mode.

```
crypto ca enroll {ca-name | system-trustpoint}
```

**Syntax Description**

- **ca-name**: Name of the CA Server.
- **system-trustpoint**: Generates the leaf certificate.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>Release 7.0.1</td>
<td>The command was modified to include the <code>system-trustpoint</code> option.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `crypto ca enroll` command to request certificates from the CA for the Rivest, Shamir, and Adelman (RSA) key pairs for the router defined by the `rakeypair, on page 212` command in trustpoint configuration mode. If no `rakeypair, on page 212` command is configured for the current trustpoint, the default RSA key pair is used for enrollment. This task is also known as enrolling with the CA. (Enrolling and obtaining certificates are two separate events, but they both occur when the `crypto ca enroll` command is issued.) When using manual enrollment, these two operations occur separately.

The router needs a signed certificate from the CA for each of the RSA key pairs on the router; if you previously generated general-purpose keys, this command obtains the one certificate corresponding to the one general-purpose RSA key pair. If you previously generated special-usage keys, this command obtains two certificates corresponding to each of the special-usage RSA key pairs.

If you already have a certificate for your keys, you are unable to configure this command; instead, you are prompted to remove the existing certificate first. (You can remove existing certificates by removing the trustpoint configuration with the `no crypto ca trustpoint` command.)

The `crypto ca enroll` command is not saved in the router configuration.

---

**Note**

The root certificate signs the leaf certificate.

**Task ID**

```
Task ID Operations ID
crypto execute
```
The following sample output is from the `crypto ca enroll` command:

Router# `crypto ca enroll msiox`
% Start certificate enrollment...
% Create a challenge password. You will need to verbally provide this password to the
% CA Administrator in order to revoke your certificate.
% For security reasons you password will not be saved in the configuration.
% Please make a note of it.
%Password
re-enter Password:
    Fingerprint: 4F35ADC9 2791997A CE211437 AFC66CF7
RP/0/RP0/CPU0:May 29 18:49:15.572 : pki_cmd: %PKI-6-LOG_INFO : certificate request pending
RP/0/RP0/CPU0:May 29 18:52:17.705 : pki_get_cert: %PKI-6-LOG_INFO : certificate is granted

This example shows how to generate a leaf certificate:

Router#`crypto ca enroll system-trustpoint`
crypto ca import

To import a certification authority (CA) certificate manually through TFTP, SFTP, or cut and paste it at the terminal, use the **crypto ca import** command in XR EXEC mode.

```
crypto ca import name certificate
```

**Syntax Description**

- **name**: Name of the certification authority (CA). This name is the same name used when the CA was declared with the `crypto ca trustpoint`, on page 186 command.
- **certificate**: None

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

```
crypto execute
```

**Examples**

The following example shows how to import a CA certificate through cut-and-paste. In this example, the certificate is myca.

```
RP/0/RP0/CPU0:router# crypto ca import myca certificate
```
**crypto ca trustpoint**

To configure a trusted point with a selected name, use the `crypto ca trustpoint` command. To unconfigure a trusted point, use the `no` form of this command in XR Config mode.

```
crypto ca trustpoint {ca-name | system-trustpoint}
```

**Syntax Description**

- `ca-name`: Name of the CA.
- `system-trustpoint`: Specifies the default system trustpoint.

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>Release 7.0.1</td>
<td>The command was modified to include the <code>system-trustpoint</code> option to specify the default system trustpoint.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `crypto ca trustpoint` command to declare a CA.

This command allows you to configure a trusted point with a selected name so that your router can verify certificates issued to peers. Your router need not enroll with the CA that issued the certificates to the peers.

The `crypto ca trustpoint` command enters trustpoint configuration mode, in which you can specify characteristics for the CA with a set of commands. See the Related Commands section for details.

**Examples**

The following example shows how to use the `crypto ca trustpoint` command to create a trustpoint:

```
Router# configure
Router(config)# crypto ca trustpoint msiox
Router(config-trustp)# sftp-password xxxxxx
Router(config-trustp)# sftp-username tmordeko
Router(config-trustp)# enrollment url sftp://192.168.254.254/tftpboot/tmordeko/CAcert
Router(config-trustp)# rsakeypair label-2
```

This example shows how to create a default system trustpoint:

```
Router# configure
```
Router(config)#crypto ca trustpoint system-trustpoint
Router(config-trustp)#commit

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca-keypair, on page 176</td>
<td>Creates the key pair for the root certificate on the router.</td>
</tr>
<tr>
<td>crl optional (trustpoint), on page 179</td>
<td>Allows the certificates of other peers to be accepted without trying to obtain the appropriate CRL.</td>
</tr>
<tr>
<td>enrollment retry count, on page 200</td>
<td>Specifies how many times a router resends a certificate request.</td>
</tr>
<tr>
<td>enrollment retry period, on page 201</td>
<td>Specifies the wait period between certificate request retries.</td>
</tr>
<tr>
<td>enrollment terminal, on page 202</td>
<td>Specifies manual cut-and-paste certificate enrollment.</td>
</tr>
<tr>
<td>enrollment url, on page 203</td>
<td>Specifies the URL of the CA.</td>
</tr>
<tr>
<td>ip-address (trustpoint), on page 205</td>
<td>Specifies a dotted IP address that is included as an unstructured address in the certificate request.</td>
</tr>
<tr>
<td>key-usage, on page 206</td>
<td>Specifies the key usage field for the self-enrollment certificate.</td>
</tr>
<tr>
<td>keypair, on page 208</td>
<td>Creates the key pair for the leaf certificate on the router.</td>
</tr>
<tr>
<td>lifetime (trustpoint), on page 209</td>
<td>Configures the lifetime for self-enrollment of certificates.</td>
</tr>
<tr>
<td>message-digest, on page 210</td>
<td>Configures the message digest hashing algorithm for the certificates.</td>
</tr>
<tr>
<td>query url, on page 211</td>
<td>Specifies the LDAP URL of the CRL distribution point. Required only if your CA supports Lightweight Directory Access Protocol (LDAP).</td>
</tr>
<tr>
<td>rsakeypair, on page 212</td>
<td>Specifies a named RSA key pair for this trustpoint.</td>
</tr>
<tr>
<td>serial-number (trustpoint), on page 213</td>
<td>Specifies a router serial number in the certificate request.</td>
</tr>
<tr>
<td>sftp-password (trustpoint), on page 214</td>
<td>Secures the FTP password.</td>
</tr>
<tr>
<td>sftp-username (trustpoint), on page 215</td>
<td>Secures the FTP username.</td>
</tr>
<tr>
<td>subject-name (trustpoint), on page 216</td>
<td>Specifies a subject name in the certificate request.</td>
</tr>
</tbody>
</table>
crypto ca trustpool import url

To manually update certificates in the trust pool if they are not current, are corrupt, or if certain certificates need to be updated, use the `crypto ca trustpool import url` command in XR EXEC mode.

```
crypto ca trustpool import url {cleanURL}  
```

**Syntax Description**

- **clean** (Optional) Manually remove all downloaded certificate authority (CA) certificates.
- **URL** Specify the URL from which the CA trust pool certificate bundle must be downloaded. This manually imports (downloads) the CA certificate bundle into the CA trust pool to update or replace the existing CA certificate bundle.

**Command Default**
The CA trust pool feature is enabled. The router uses the built-in CA certificate bundle in the CA trust pool which is updated automatically from Cisco.

**Command Modes**
XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
The CA trust pool feature is enabled by default and uses the built-in CA certificate bundle in the trust pool, which receives automatic updates from Cisco. Use the `crypto ca trustpool import url` manually update certificates in the trust pool if they are not current, are corrupt, or if certain certificates need to be updated.

**Example**
This example shows how to run the command to manually update certificates in the trust pool if they are not current, are corrupt, or if certain certificates need to be updated.

```
RP/0/RP0/CPU0:router#crypto ca trustpool import url
```
crypto ca trustpool policy

To configure certificate authority (CA) trust pool policy, use the **crypto ca trustpool policy** command in XR Config mode.

```bash
crypto ca trustpool policy {cabundle url url | crl optional | description line}
```

**Syntax Description**
- `cabundle url URL` Configures the URL from which the CA trust pool bundle is downloaded.
- `crl optional` To specify the certificate revocation list (CRL) query for the CA trust pool, use the `crl` command in ca-trustpool configuration mode. By default, the router enforces a check of the revocation status of the certificate by querying the certificate revocation list (CRL). Setting this to `optional` disables revocation checking when the trust pool policy is in use.
- `description line` Indicates the description for the trust pool policy.

**Command Default**
The default CA trust pool policy is used.

**Command Modes**
XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
The **crypto ca trustpool policy** command enters ca-trustpool configuration mode, where commands can be accessed to configure certificate authority (CA) trustpool policy parameters.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>READ, WRITE</td>
</tr>
</tbody>
</table>

**Example**

This example shows you how to disable certificate revocation checks when the trust pool policy is in use.

```bash
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)#crypto ca trustpool policy
RP/0/RP0/CPU0:router(config-trustpool)#crl optional
```
crypto key generate dsa

To generate Digital Signature Algorithm (DSA) key pairs, use the **crypto key generate dsa** command in XR EXEC mode.

**crypto key generate dsa**

**Syntax Description**

- **system-enroll-key** Specifies key pair generation for the leaf certificate.
- **system-root-key** Specifies key pair generation for the root certificate.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>Release 7.0.1</td>
<td>The command was modified to include system-enroll-key and system-root-key options for the key pair generation of leaf and root certificates.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the **crypto key generate dsa** command to generate DSA key pairs for your router.

DSA keys are generated in pairs—one public DSA key and one private DSA key.

If your router already has DSA keys when you issue this command, you are warned and prompted to replace the existing keys with new keys.

To remove the DSA key generated, use the **crypto key zeroize dsa** command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>execute</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to generate a 512-bit DSA key:

```
RP/0/RP0/CPU0:router# crypto key generate dsa
The name for the keys will be: the_default
Choose the size of your DSA key modulus. Modulus size can be 512, 768, or 1024 bits.
Choosing a key modulus
How many bits in the modulus [1024]: 512
Generating DSA keys...
Done w/ crypto generate keypair [OK]
```

This example shows how to generate a DSA key pair for the root certificate:
Router#crypto key generate dsa system-root-key

This example shows how to generate a DSA key pair for the leaf certificate:

Router#crypto key generate dsa system-enroll-key
crypto key generate ecdsa

To generate an Elliptic Curve Digital Signature Algorithm (ECDSA) key pair, use the `crypto key generate ecdsa` command.

```
crypto key generate ecdsa [{nistp256|nistp384|nistp521}] [{system-enroll-key | system-root-key}]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nistp256</td>
<td>Generates an ECDSA key of curve type nistp256, with key size 256 bits.</td>
</tr>
<tr>
<td>nistp384</td>
<td>Generates an ECDSA key of curve type nistp384, with key size 384 bits.</td>
</tr>
<tr>
<td>nistp521</td>
<td>Generates an ECDSA key of curve type nistp521, with key size 521 bits.</td>
</tr>
<tr>
<td>system-enroll-key</td>
<td>Specifies key pair generation for the leaf certificate.</td>
</tr>
<tr>
<td>system-root-key</td>
<td>Specifies key pair generation for the root certificate.</td>
</tr>
</tbody>
</table>

**Command Default**

None

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>7.0.1</td>
<td>The command was modified to include <code>system-enroll-key</code> and <code>system-root-key</code> options for the key pair generation of leaf and root certificates.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

To remove an ECDSA key, use the `crypto key zeroize ecdsa` command.

**Example**

The following example shows how to generate a ECDSA key pair:

```
Router# crypto key generate ecdsa nistp384
Wed Mar 28 12:53:57.355 UTC
% You already have keys defined for the_default
Do you really want to replace them? [yes/no]: yes
Generating ECDSA keys ... Done w/ crypto generate ECDSA keypair
[OK]
```

This example shows how to generate a ECDSA key pair for the root certificate:
Router#crypto key generate ecdsa system-root-key

This example shows how to generate a ECDSA key pair for the leaf certificate:

Router#crypto key generate dsa system-enroll-key
crypto key generate rsa

To generate a Rivest, Shamir, and Adelman (RSA) key pair, use the **crypto key generate rsa** command in XR EXEC mode.

```
crypto key generate rsa [ { usage-keys | general-keys | system-enroll-key | system-root-key } ]
[ keypair-label ]
```

**Syntax Description**

- `usage-keys` *(Optional)*: Generates separate RSA key pairs for signing and encryption.
- `general-keys` *(Optional)*: Generates a general-purpose RSA key pair for signing and encryption.
- `keypair-label` *(Optional)*: RSA key pair label that names the RSA key pairs.
- `system-enroll-key`: Specifies key pair generation for the leaf certificate.
- `system-root-key`: Specifies key pair generation for the root certificate.

**Command Default**

RSA key pairs do not exist. If the `usage-keys` keyword is not used, general-purpose keys are generated. If no RSA label is specified, the key is generated as the default RSA key.

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>Release 7.0.1</td>
<td>The command was modified to include <code>system-enroll-key</code> and <code>system-root-key</code> options for the key pair generation of leaf and root certificates.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the **crypto key generate rsa** command to generate RSA key pairs for your router.

RSA keys are generated in pairs—one public RSA key and one private RSA key.

If your router already has RSA keys when you issue this command, you are warned and prompted to replace the existing keys with new keys. The keys generated by this command are saved in the secure NVRAM (which is not displayed to the user or backed up to another device).

To remove an RSA key, use the **crypto key zeroize rsa** command.

**Examples**

The following example shows how to generate an RSA key pair:
Router# crypto key generate rsa

The name for the keys will be: the_default

Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus[1024]: <return>
Router#

This example shows how to generate an RSA key pair for the root certificate:

Router# crypto key generate rsa system-root-key

This example shows how to generate an RSA key pair for the leaf certificate:

Router# crypto key generate rsa system-enroll-key
crypto key import authentication rsa

To import a public key using the Rivest, Shamir, and Adelman (RSA) method, use the **crypto key import authentication rsa** command in XR EXEC mode.

\[ \text{crypto key import authentication rsa } path \]

**Syntax Description**

- **path** *(Optional)*: This denotes the path to the RSA public key file.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

1. Use ssh-keygen generation mechanism to generate keys using either a LINUX or UNIX client. This creates two keys: one public and one private.

2. Remove the comment and other header tag from the keys, except the base64encoded text.

3. Decode the base64encoded text, and use the for authentication.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>execute</td>
</tr>
</tbody>
</table>

**Examples**

The following example displays how to import a public key:

```
RP/0/RP0/CPU0:router#crypto key import authentication rsa
```
crypto key zeroize dsa

To delete the Digital Signature Algorithm (DSA) key pair from your router, use the **crypto key zeroize dsa** command in XR EXEC mode.

**crypto key zeroize dsa**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>This command has no keywords or arguments.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Default</td>
<td>None</td>
</tr>
<tr>
<td>Command Modes</td>
<td>XR EXEC mode</td>
</tr>
</tbody>
</table>

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the **crypto key zeroize dsa** command to delete the DSA key pair that was previously generated by your router.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>execute</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to delete DSA keys from your router:

```
RP/0/RP0/CPU0:router# crypto key zeroize dsa
% Keys to be removed are named the_default
Do you really want to remove these keys? [yes/no]: yes
```
crypto key zeroize rsa

to delete all Rivest, Shamir, and Adelman (RSA) keys from the router, use the crypto key zeroize rsa command in XR EXEC mode.

crypto key zeroize rsa [keypair-label]

Syntax Description

keypair-label (Optional) Names the RSA key pair to be removed.

Command Default

If the key pair label is not specified, the default RSA key pair is removed.

Command Modes

XR EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines

Use the crypto key zeroize rsa command to delete all RSA keys that were previously generated by the router. After issuing this command, you must perform two additional tasks:

- Ask the certification authority (CA) administrator to revoke the certificates for the router at the CA; you must supply the challenge password you created when you originally obtained the router certificates with the crypto ca enroll, on page 183 command CA.
- Manually remove the certificates from the configuration using the clear crypto ca certificates command.

Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>execute</td>
</tr>
</tbody>
</table>

Examples

The following example shows how to delete the general-purpose RSA key pair that was previously generated:

```
RP/0/RP0/CPU0:router# crypto key zeroize rsa key1
% Keys to be removed are named key1
Do you really want to remove these keys? [yes/no]: yes
```
**description (trustpoint)**

To create a description of a trustpoint, use the `description` command in trustpoint configuration mode. To delete a trustpoint description, use the `no` form of this command.

```
description string
no description
```

**Syntax Description**

`string`  Character string describing the trustpoint.

**Command Default**

The default description is blank.

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `description` command in the trustpoint configuration mode to create a description for a trustpoint.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to create a trustpoint description:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0/CPU0:router(config-trustp)# description this is the primary trustpoint
```
enrollment retry count

To specify the number of times a router resends a certificate request to a certification authority (CA), use the `enrollment retry count` command in trustpoint configuration mode. To reset the retry count to the default, use the `no` form of this command.

```
enrollment retry count number
no enrollment retry count number
```

**Syntax Description**

- `number`: Number of times the router resends a certificate request when the router does not receive a certificate from the previous request. The range is from 1 to 100.

**Command Default**

If no retry count is specified, the default value is 10.

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

After requesting a certificate, the router waits to receive a certificate from the CA. If the router does not receive a certificate within a specified time (the retry period), the router sends another certificate request. The router continues to send requests until it receives a valid certificate, the CA returns an enrollment error, or the configured number of retries (the retry count) is exceeded.

To reset the retry count to the default of 10, use the `no` form of this command. Setting the retry count to 0 indicates an infinite number of retries. The router sends the CA certificate requests until a valid certificate is received (there is no limit to the number of retries).

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to declare a CA, change the retry period to 10 minutes, and change the retry count to 60 retries. The router resends the certificate request every 10 minutes until receipt of the certificate or approximately 10 hours pass since the original request was sent, whichever occurs first (10 minutes x 60 tries = 600 minutes = 10 hours).

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0/CPU0:router(config-trustp)# enrollment url http://ca_server
RP/0/RP0/CPU0:router(config-trustp)# enrollment retry period 10
RP/0/RP0/CPU0:router(config-trustp)# enrollment retry count 60
```
enrollment retry period

To specify the wait period between certificate request retries, use the enrollment retry period command in trustpoint configuration mode. To reset the retry period to the default of 1 minute, use the no form of this command.

enrollment retry period minutes
no enrollment retry period minutes

Syntax Description

- **minutes**: Period (in minutes) between certificate requests issued to a certification authority (CA) from the router. The range is from 1 to 60 minutes.

Command Default

- **minutes**: 1

Command Modes

- Trustpoint configuration

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines

After requesting a certificate, the router waits to receive a certificate from the CA. If the router does not receive a certificate within a specified time (the retry period), the router sends another certificate request. The router continues to send requests until it receives a valid certificate, the CA returns an enrollment error, or the configured number of retries (the retry count) is exceeded.

The router sends the CA another certificate request every minute until a valid certificate is received. (By default, the router sends ten requests, but you can change the number of permitted retries with the enrollment retry count command.)

Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

Examples

The following example shows how to declare a CA and change the retry period to 5 minutes:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0/CPU0:router(config-trustp)# enrollment retry period 5
```
**enrollment terminal**

To specify manual cut-and-paste certificate enrollment, use the `enrollment terminal` command in trustpoint configuration mode. To delete a current enrollment request, use the `no` form of this command.

```
enrollment terminal
no enrollment terminal
```

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

You can manually cut and paste certificate requests and certificates when you do not have a network connection between the router and certification authority (CA). When the `enrollment terminal` command is enabled, the router displays the certificate request on the console terminal, which allows you to enter the issued certificate on the terminal.

**Task ID**

<table>
<thead>
<tr>
<th>Task</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>read, write</td>
</tr>
<tr>
<td>crypto</td>
<td></td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to manually specify certificate enrollment through cut-and-paste. In this example, the CA trustpoint is myca.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0/CPU0:router(config-trustp)# enrollment terminal
```
enrollment url

To specify the certification authority (CA) location by naming the CA URL, use the enrollment url command in trustpoint configuration mode. To remove the CA URL from the configuration, use the no form of this command.

enrollment url CA-URL
no enrollment url CA-URL

Syntax Description

**CA-URL**  URL of the CA server. The URL string must start with http://CA_name, where CA_name is the host Domain Name System (DNS) name or IP address of the CA (for example, http://ca-server).

If the CA cgi-bin script location is not /cgi-bin/pkcclient.exe at the CA (the default CA cgi-bin script location), you must also include the nonstandard script location in the URL, in the form of http://CA-name/script-location, where script-location is the full path to the CA scripts.

Command Default

None

Command Modes

Trustpoint configuration

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines

Use the enrollment url command to specify the CA URL. This command is required when you declare a CA with the crypto ca trustpoint command. The URL must include the CA script location if the CA scripts are not loaded into the default cgi-bin script location. The CA administrator should be able to tell you where the CA scripts are located.

This table lists the available enrollment methods.

**Table 9: Certificate Enrollment Methods**

<table>
<thead>
<tr>
<th>Enrollment Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFTP</td>
<td>Enroll through SFTP: file system</td>
</tr>
<tr>
<td>TFTP(^1)</td>
<td>Enroll through TFTP: file system</td>
</tr>
</tbody>
</table>

\(^1\) If you are using TFTP for enrollment, the URL must be in the form tftp://certserver/file_specification. (The file specification is optional.)

TFTP enrollment sends the enrollment request and retrieves the certificate of the CA and the certificate of the router. If the file specification is included in the URL, the router appends an extension to the file specification.

To change the CA URL, repeat the enrollment url command to overwrite the previous URL.
The following example shows the absolute minimum configuration required to declare a CA:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)#
crypto ca trustpoint myca
RP/0/RP0/CPU0:router(config-trustp)#
enrollment url http://ca.domain.com/certsrv/mscep/mscep.dll
```
ip-address (trustpoint)

To specify a dotted IP address that is included as an unstructured address in the certificate request, use the `ip-address` command in trustpoint configuration mode. To restore the default behavior, use the `no` form of this command.

```
ip-address  {ip-address | none}
no ip-address  {ip-address | none}
```

**Syntax Description**
- `ip-address`  Dotted IP address that is included in the certificate request.
- `none`  Specifies that an IP address is not included in the certificate request.

**Command Default**
You are prompted for the IP address during certificate enrollment.

**Command Modes**
Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
Use the `ip-address` command to include the IP address of the specified interface in the certificate request or to specify that an IP address should not be included in the certificate request.

**Task ID**
- Task ID: crypto
  - Operations: read, write

**Examples**
The following example shows how to include the IP address of the Ethernet-0 interface in the certificate request for the trustpoint frog:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint frog
RP/0/RP0/CPU0:router(config-trustp)# enrollment url http://frog.phoobin.com
RP/0/RP0/CPU0:router(config-trustp)# subject-name OU=Spiral Dept., O=tiedye.com
RP/0/RP0/CPU0:router(config-trustp)# ip-address 172.19.72.120
```

The following example shows that an IP address is not to be included in the certificate request:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0/CPU0:router(config-trustp)# enrollment url http://10.3.0.7:80
RP/0/RP0/CPU0:router(config-trustp)# subject-name CN=subject1, OU=PKI, O=Cisco Systems, C=US
RP/0/RP0/CPU0:router(config-trustp)# ip-address none
```
key-usage

To specify the key usage field for the self-enrollment certificate, use the key-usage command in trustpoint configuration mode. To remove this configuration, use the no form of this command.

```
key-usage {ca-certificate {crlsign | digitalsignature | keycertsign | nonrepudiation} | certificate {dataencipherment | digitalsignature | keyagreement | keyencipherment | nonrepudiation}}
```

**Syntax Description**

- **ca-certificate**: Specifies the key usage field for the CA certificate.
- **certificate**: Specifies the key usage field for the leaf certificate.
- **crlsign**: Asserts **cRLSign** (bit 6) for the key usage field to verify signatures on certificate revocation list (CRL).
- **digitalsignature**: Asserts **digitalSignature** (bit 0) for the key usage field.
  
  This is used when the subject public key is used with a digital signature mechanism to support security services other than certificate signing (bit 5), or CRL signing (bit 6).
- **keycertsign**: Asserts **keyCertSign** (bit 5) for the key usage field when the subject public key is used for verifying a signature on public key certificates.
- **nonrepudiation**: Asserts **nonRepudiation** (bit 1) for the key usage field when the subject public key is used to verify digital signatures that is used to provide a non-repudiation service.
- **dataencipherment**: Asserts **dataEncipherment** (bit 3) for the key usage field when the subject public key is used for enciphering user data, other than cryptographic keys.
- **keyagreement**: Asserts **keyAgreement** (bit 4) for the key usage field when the subject public key is used for key agreement.
- **keyencipherment**: Asserts **keyEncipherment** (bit 2) for the key usage field when the subject public key is used for key transport.

**Command Default**

None

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.
Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

Examples

This example shows how to specify the key usage field for the self-enrollment certificate:

Router(config)#crypto ca trustpoint system-trustpoint
Router(config-trustp)#key-usage certificate digitalsignature keyagreement dataencipherment
Router(config-trustp)#commit
keypair

To create the key pair for the leaf certificate on the router, use the `keypair` command in trustpoint configuration mode. To remove this configuration, use the `no` form of this command.

```
keypair  {dsa | ecdsanistp256 | ecdsanistp384 | ecdsanistp521 | rsa}  key-pair-label
```

**Syntax Description**

- `key-pair-label`  Specifies the key pair label for the respective key signature algorithm (DSA, ECDSA or RSA).

**Command Default**

None

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

- **crypto**  read, write

**Examples**

This example shows how to create the key pair for the leaf certificate on the router:

```
Router# configure
Router(config)# crypto ca trustpoint system-trustpoint
Router(config-trustp)# keypair rsa system-enroll-key
Router(config-trustp)# commit
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ca-keypair</code>, on page 176</td>
<td>Creates the key pair for the root certificate on the router.</td>
</tr>
</tbody>
</table>
To configure the lifetime for self-enrollment of certificates, use the `lifetime` command in trustpoint configuration mode. To remove this configuration, use the `no` form of this command.

```
lifetime  {ca-certificate | certificate}  validity
```

**Syntax Description**

- `ca-certificate`: Configures the lifetime for self-enrollment of CA certificate.  

- `validity`: Specifies the validity for the certificates, in days.  
  The range is from 30 to 5474 days.

**Command Default**

None

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
</tr>
<tr>
<td>read,</td>
</tr>
<tr>
<td>write</td>
</tr>
</tbody>
</table>

**Examples**

This example shows how to configure the lifetime for self-enrollment of CA certificate:

```
Router# configure
Router(config)# crypto ca trustpoint system-trustpoint
Router(config-trustp)# lifetime ca-certificate 30
Router(config-trustp)# commit
```
message-digest

To configure the message digest hashing algorithm for the certificates, use the `message-digest` command in trustpoint configuration mode. To remove this configuration, use the `no` form of this command.

```
message-digest {md5 | sha1 | sha256 | sha384 | sha512}
```

**Syntax Description**

- **md5**: Specifies MD5 as the message digest hashing algorithm for the certificate.
- **sha1**: Specifies SHA1 as the message digest hashing algorithm for the certificate.
- **sha256**: Specifies SHA256 as the message digest hashing algorithm for the certificate.
- **sha384**: Specifies SHA384 as the message digest hashing algorithm for the certificate.
- **sha512**: Specifies SHA512 as the message digest hashing algorithm for the certificate.

**Command Default**

None

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

- crypto: read, write

**Examples**

This example shows how to specify SHA256 as the message digest hashing algorithm for the certificate:

```
Router(config)#crypto ca trustpoint system-trustpoint
Router(config-trustp)#message-digest sha256
Router(config-trustp)#commit
```
query url

To specify Lightweight Directory Access Protocol (LDAP) protocol support, use the `query url` command in trustpoint configuration mode. To remove the query URL from the configuration, use the `no` form of this command.

```
query url  LDAP-URL
no query url  LDAP-URL
```

**Syntax Description**

`LDAP-URL` URL of the LDAP server (for example, ldap://another-server).

This URL must be in the form of ldap://server-name where server-name is the host Domain Name System (DNS) name or IP address of the LDAP server.

**Command Default**

The URL provided in the router certificate’s CRLDistributionPoint extension is used.

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

LDAP is a query protocol used when the router retrieves the Certificate Revocation List (CRL). The certification authority (CA) administrator should be able to tell you whether the CA supports LDAP; if the CA supports LDAP, the CA administrator can tell you the LDAP location where certificates and certificate revocation lists should be retrieved.

To change the query URL, repeat the `query url` command to overwrite the previous URL.

**Task ID**

```
Task ID  Operations
read,    write
```

**Examples**

The following example shows the configuration required to declare a CA when the CA supports LDAP:

```
RP/0/RP0/CPU0:router(config-trustp)# query url ldap://my-ldap.domain.com
```
To specify a named Rivest, Shamir, and Adelman (RSA) key pair for this trustpoint, use the `rsakeypair` command in trustpoint configuration mode. To reset the RSA key pair to the default, use the `no` form of this command.

```
rsakeypair  keypair-label
no  rsakeypair  keypair-label
```

### Syntax Description

- `keypair-label` RSA key pair label that names the RSA key pairs.

### Command Default

If the RSA key pair is not specified, the default RSA key is used for this trustpoint.

### Command Modes

Trustpoint configuration

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

Use the `rsakeypair` command to specify a named RSA key pair generated using the `crypto key generate rsa` command for this trustpoint.

### Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

### Examples

The following example shows how to specify the named RSA key pair key1 for the trustpoint myca:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0/CPU0:router(config-trustp)# rsakeypair key1
```
serial-number (trustpoint)

To specify whether the router serial number should be included in the certificate request, use the `serial-number` command in trustpoint configuration mode. To restore the default behavior, use the `no` form of this command.

```
serial-number [none]
no serial-number
```

**Syntax Description**

- `none` (Optional) Specifies that a serial number is not included in the certificate request.

**Command Default**

You are prompted for the serial number during certificate enrollment.

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Before you can use the `serial-number` command, you must enable the `crypto ca trustpoint` command, which declares the certification authority (CA) that your router should use and enters trustpoint configuration mode.

Use this command to specify the router serial number in the certificate request, or use the `none` keyword to specify that a serial number should not be included in the certificate request.

**Examples**

The following example shows how to omit a serial number from the root certificate request:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint root
RP/0/RP0/CPU0:router(config-trustp)# enrollment url http://10.3.0.7:80
RP/0/RP0/CPU0:router(config-trustp)# ip-address none
RP/0/RP0/CPU0:router(config-trustp)# serial-number none
RP/0/RP0/CPU0:router(config-trustp)# subject-name ON=Jack, OU=PKI, O=Cisco Systems, C=US
```
sftp-password (trustpoint)

To secure the FTP password, use the **sftp-password** command in trustpoint configuration mode. To disable this feature, use the **no** form of this command.

```
sftp-password {clear text | clear text | password encrypted string}
no sftp-password {clear text | clear text | password encrypted string}
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>clear text</strong></td>
<td>Clear text password and is encrypted only for display purposes.</td>
</tr>
<tr>
<td><strong>password encrypted string</strong></td>
<td>Enters the password in an encrypted form.</td>
</tr>
</tbody>
</table>

**Command Default**

The **clear text** argument is the default behavior.

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Passwords are stored in encrypted form and not as plain text. The command-line interface (CLI) contains the provisioning (for example, clear and encrypted) to specify the password input.

The username and password are required as part of the SFTP protocol. If you specify the URL that begins with the prefix (sftp://), you must configure the parameters for the **sftp-password** command under the trustpoint. Otherwise, the certificate from the SFTP server, which is used for manual certificate enrollment, cannot be retrieved.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to secure the FTP password in an encrypted form:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint msiox
RP/0/RP0/CPU0:router(config-trustp)# sftp-password password xxxxxx
```
**sftp-username (trustpoint)**

To secure the FTP username, use the `sftp-username` command in trustpoint configuration mode. To disable this feature, use the `no` form of this command.

```
sftp-username  username
no sftp-username  username
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>username</code></td>
<td>Name of the user.</td>
</tr>
</tbody>
</table>

**Command Default**

None

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `sftp-username` command is used only if the URL has `(sftp://)` in the prefix. If `(sftp://)` is not specified in the prefix, the manual certificate enrollment using SFTP fails.

**Examples**

The following example shows how to secure the FTP username:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint msiox
RP/0/RP0/CPU0:router(config-trustp)# sftp-username tmordeko
```
subject-name (trustpoint)

To specify the subject name in the certificate request, use the `subject-name` command in trustpoint configuration mode. To clear any subject name from the configuration, use the `no` form of this command.

```
subject-name [ca-certificate] subject-name
```

**Syntax Description**

- **ca-certificate** (Optional) Specifies the subject name for the CA certificate for self-enrollment.
- **subject-name** (Optional) Specifies the subject name used in the certificate request.

**Command Default**

If the `subject-name` argument is not specified, the fully qualified domain name (FQDN), which is the default subject name, is used.

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>Release 7.0.1</td>
<td>The command was modified to include the <code>ca-certificate</code> option.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Before you can use the `subject-name` command, you must enable the `crypto ca trustpoint` command, which declares the certification authority (CA) that your router should use and enters trustpoint configuration mode.

The `subject-name` command is an attribute that can be set for automatic enrollment; thus, issuing this command prevents you from being prompted for a subject name during enrollment.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to specify the subject name for the frog certificate:

```
Router# configure
Router(config)# crypto ca trustpoint frog
Router(config-trustp)# enrollment url http://frog.phoobin.com
Router(config-trustp)# subject-name OU=Spiral Dept., O=tiedye.com
Router(config-trustp)# ip-address 172.19.72.120
```

This example shows how to specify the subject name for the CA certificate for self-enrollment.

```
Router# configure
Router(config)# crypto ca trustpoint system-trustpoint
Router(config-trustp)#subject-name ca-certificate CN=labuser-ca,C=US,ST=CA,L=San Jose,O=cisco systems,O=ASR
```
Router(config-trustp)#commit
show crypto ca certificates

To display information about your certificate and the certification authority (CA) certificate, use the `show crypto ca certificates` command in XR EXEC mode.

**Syntax Description**
- This command has no keywords or arguments.

**Command Default**
- None

**Command Modes**
- XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `show crypto ca certificates` command to display information about the following certificates:

- Your certificate, if you have requested one from the CA (see the `crypto ca enroll` command).
- CA certificate, if you have received the certificate (see the `crypto ca authenticate` command).

**Examples**

The following sample output is from the `show crypto ca certificates` command:

```plaintext
RP/0/RP0/CPU0:router# show crypto ca certificates
Trustpoint : msiox
---------------------------------------------------
CAa certificate
Subject:
  Name: CA2
  CN= CA2
Issued By :
  cn=CA2
Validity Start : 07:51:51 UTC Wed Jul 06 2005
Validity End : 08:00:43 UTC Tue Jul 06 2010
CRL Distribution Point
  http://10.56.8.236/CertEnroll/CA2.crl
Router certificate
Status : Available
Key usage : Signature
Serial Number : 38:6B:C6:B8:00:04:00:00:01:45
Subject:
  Name: tdlr533.cisco.com
  IP Address: 3.1.53.3
  Serial Number: 8cd96b64
Issued By :
  cn=CA2
```
Validity Start : 08:30:03 UTC Mon Apr 10 2006
Validity End   : 08:40:03 UTC Tue Apr 10 2007
CRL Distribution Point
   http://10.56.8.236/CertEnroll/CA2.crl
Associated Trustpoint: MS-IOX
Router certificate
Status        : Available
Key usage     : Encryption
Serial Number : 38:6D:2B:A7:00:04:00:00:01:46
Subject:
   Name: tdlr533.cisco.com
   IP Address: 3.1.53.3
   Serial Number: 8cd96b64
Issued By:
   cn=CA2
Validity Start : 08:31:34 UTC Mon Apr 10 2006
Validity End   : 08:41:34 UTC Tue Apr 10 2007
CRL Distribution Point
   http://10.56.8.236/CertEnroll/CA2.crl
Associated Trustpoint: msiox
show crypto ca crls

To display information about the local cache Certificate Revocation List (CRL), use the `show crypto ca crls` command in XR EXEC mode.

**show crypto ca crls**

**Syntax Description**
This command has no keywords or arguments.

**Command Default**
None

**Command Modes**
XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**
The following sample output is from the `show crypto ca crls` command:

```
RP/0/RP0/CPU0:router# show crypto ca crls
CRL Entry
===============================================
Issuer : cn=xyz-w2k-root,ou=HFR,o=Cisco System,l=San Jose,st=CA,c=US
Last Update : [UTC] Thu Jan 10 01:01:14 2002
CRL Distribution Point :
  http://xyz-w2k.cisco.com/CertEnroll/xyz-w2k-root.crl
```
**show crypto ca trustpool policy**

To display the CA trust pool certificates of the router in a verbose format use the `show crypto ca trustpool policy` command in XR EXEC mode.

**show crypto ca trustpool policy**

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

No default behavior or values.

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the command to display the CA trust pool certificates of the router in a verbose format.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read</td>
</tr>
</tbody>
</table>

**Example**

This example shows you how to run the command to view details of your CA certificate trust pool policy.

```
RP/0/RP0/CPU0:router# show crypto ca trustpool policy

Trustpool Policy

Trustpool CA certificates will expire [UTC] Thu Sep 30 14:01:15 2021
CA Bundle Location: http://cisco.com/security/pki/trs/ios.p7b
```
show crypto key mypubkey dsa

To display the Directory System Agent (DSA) public keys for your router, use the `show crypto key mypubkey dsa` command in XR EXEC mode.

**show crypto key mypubkey dsa**

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**

The following sample output is from the `show crypto key mypubkey dsa` command:

```
RP/0/RP0/CPU0:router# show crypto key mypubkey dsa

Key label: mykey
Type : RSA General purpose
Size : 1024
Created : 17:33:23 UTC Thu Sep 18 2003
Data :
3081f230 81aa0605 2b0e0302 0c3081a0 02020200 024100c8 a36b6179 56b8d620
1f77595c 32ef3004 577a9f79 0a8abda4 89fb969d 35c04e7e 5491ed4e 120c657c
610576e5 841696b6 0948846c c92f56e5 b4921458 70fc4902 1500abd1 5c0d63d3
8e082ba9 f16030c5 aa05d1a dfe50240 73f661ea 9f579e77 b413bc4 9047b4f2
10a1cfcb 14d98b57 3e0bba97 9b5120ad f52bbdc7 1b5b6345 8c5b88s5 92b6c9df
7dc27768 fd296844 42024945 5e86c81a 03430002 4071b49e f80f9e4b af2b62e7
aa817460 87ef503 c668ad8c d606050b 225cc277 7c0a0974 8072d7d7 2add8e42
329fe896 ab01ed1 3a414254 6935fdca 0043ba4f 66
```
show crypto key mypubkey rsa

To display the Rivest, Shamir, and Adelman (RSA) public keys for your router, use the show crypto key mypubkey rsa command in XR EXEC mode.

Syntax Description

This command has no keywords or arguments.

Command Default

None

Command Modes

XR EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines

No specific guidelines impact the use of this command.

Task ID

<table>
<thead>
<tr>
<th>Task</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read</td>
</tr>
</tbody>
</table>

Examples

The following is sample output from the show crypto key mypubkey rsa command:

```
RP/0/RP0/CPU0:router# show crypto key mypubkey rsa

Key label: mykey
Type : RSA General purpose
Size : 1024
Created : 07:46:15 UTC Fri Mar 17 2006
Data : 30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00CF8C9DF
5BFCA055 DA4D164D F6EDB78B 926B1IDDE 0383027F BA71BCC6 9D5592C4 5BA8670E
35CD19B7 1C973A46 62CC5F8C 82BD596C F292410F 8E83B753 4BA71BAC 41AB6B6D
F34A2499 EDE11639 F88B4210 B2A0CF5F DD678C36 0D8B7DE1 A2AB5122 9ED947D5
76CF5BBD D9A2039F D02841B0 7F8BFF97 C080B791 10A9ED41 00FB6F40 95020301
0001

Key label: the_default
Type : RSA General purpose
Size : 512
Created : 07:46:15 UTC Fri Mar 17 2006
Data : 305C300D 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00CF8C9D
5BFCA055 DA4D164D F6EDB78B 926B1IDDE 0383027F BA71BCC6 9D5592C4 5BA8670E
35CD19B7 1C973A46 62CC5F8C 82BD596C F292410F 8E83B753 4BA71BAC 41AB6B6D
F34A2499 EDE11639 F88B4210 B2A0CF5F DD678C36 0D8B7DE1 A2AB5122 9ED947D5
76CF5BBD D9A2039F D02841B0 7F8BFF97 C080B791 10A9ED41 00FB6F40 95020301
0001
```
show platform security integrity dossier

To collect the data from various IOS XR applications, use the `show platform security integrity dossier` command in XR EXEC mode.

```
show platform security integrity dossier [include {packages | reboot-history | rollback-history | running-config | system-integrity-snapshot | system-inventory}] [nonce nonce-value]
```

**Syntax Description**

- **packages**: Displays active package(s) installed.
- **reboot-history**: Displays reboot history of the node.
- **rollback-history**: Displays rollback history of the node.
- **running-config**: Displays the currently committed running configuration on the node, as displayed by `show running configuration` command.
- **system-integrity-snapshot**: Displays the system integrity snapshot.
- **system-inventory**: Displays the system inventory.
- **nonce**: Specifies the nonce to generate the signature.
- **nonce-value**: Specifies the nonce value in hexadecimal string format.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The output of this command is displayed in JSON format.

**Task ID**

<table>
<thead>
<tr>
<th>Options</th>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>packages</td>
<td>pkg-mgmt</td>
<td>read</td>
</tr>
<tr>
<td>reboot-history</td>
<td>system</td>
<td>read</td>
</tr>
<tr>
<td>rollback-history</td>
<td>config-services</td>
<td>read</td>
</tr>
<tr>
<td>running-config</td>
<td>NA (available to all users)</td>
<td>read</td>
</tr>
<tr>
<td>system-integrity-snapshot</td>
<td>basic-services</td>
<td>read</td>
</tr>
<tr>
<td>system-inventory</td>
<td>sysmgr</td>
<td>read</td>
</tr>
</tbody>
</table>
Examples

This example shows the usage of **show platform security integrity dossier** command with various selectors:

```
Router#show platform security integrity dossier include packages reboot-history rollback-history system-integrity-snapshot system-inventory nonce 1580 | utility sign nonce 1580 include-certificate
```
utility sign

To sign the command output with the enrollment key to verify its data integrity and authenticity, use the utility sign command along with any of the Cisco IOS XR commands.

```
utility sign [{include-certificate | nonce nonce-value}]
```

**Syntax Description**

- `include-certificate`: Includes the certificate of the signer.
- `nonce`: Indicates the nonce to generate the signature.
- `nonce-value`: Specifies the nonce value in hexadecimal string format.

**Command Default**

None

**Command Modes**

Any IOS XR command configuration mode.

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>execute</td>
</tr>
</tbody>
</table>

**Examples**

This example shows how to add a signature to the command output data to verify its data integrity and authenticity:

```
Router# show version | utility sign nonce 1234 include-certificate
```
Secure Shell Commands

This module describes the Cisco IOS XR software commands used to configure Secure Shell (SSH).

Note

All commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router that is introduced from Cisco IOS XR Release 6.3.2. References to earlier releases in Command History tables apply to only the Cisco NCS 5500 Series Router.

Note

- Starting with Cisco IOS XR Release 6.6.25, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 560 Series Routers.

- Starting with Cisco IOS XR Release 6.3.2, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router.

- References to releases before Cisco IOS XR Release 6.3.2 apply to only the Cisco NCS 5500 Series Router.

- Cisco IOS XR Software Release 7.0.1 specific updates are not applicable for the following variants of Cisco NCS 540 Series Routers:
  - N540-28Z4C-SYS-A
  - N540-28Z4C-SYS-D
  - N540X-16Z4G8Q2C-A
  - N540X-16Z4G8Q2C-D
  - N540-12Z20G-SYS-A
  - N540-12Z20G-SYS-D
  - N540X-12Z16G-SYS-A
  - N540X-12Z16G-SYS-D

For detailed information about SSH concepts, configuration tasks, and examples, see the Implementing Secure Shell chapter in the System Security Configuration Guide for Cisco NCS 5500 Series Routers.
Currently, only default VRF is supported. VPNv4, VPNv6 and VPN routing and forwarding (VRF) address families will be supported in a future release.

- clear ssh, on page 229
- netconf-yang agent ssh, on page 231
- sftp, on page 232
- sftp (Interactive Mode), on page 235
- show ssh, on page 238
- show ssh history, on page 240
- show ssh history details, on page 242
- show ssh session details, on page 244
- ssh, on page 246
- ssh algorithms cipher, on page 248
- ssh client enable cipher, on page 249
- ssh client knownhost, on page 251
- ssh client source-interface, on page 252
- ssh client vrf, on page 254
- ssh server, on page 255
- ssh server algorithms host-key, on page 256
- ssh server disable hmac, on page 257
- ssh server enable cipher, on page 258
- ssh server logging, on page 259
- ssh server rate-limit, on page 260
- ssh server session-limit, on page 261
- ssh server v2, on page 262
- ssh server vrf, on page 263
- ssh server netconf, on page 265
- show tech-support ssh, on page 266
- ssh timeout, on page 268
clear ssh

To terminate an incoming or outgoing Secure Shell (SSH) connection, use the clear ssh command.

clear ssh {session-id | outgoing session-id}

Syntax Description

| Session ID number of an incoming connection as displayed in the show ssh command output. Range is from 0 to 1024. |
| session-id |

| Specifies the session ID number of an outgoing connection as displayed in the show ssh command output. Range is from 1 to 10. |
| outgoing session-id |

Command Default

None

Command Modes

XR EXEC mode

Command History

Release 6.0 This command was introduced.

Usage Guidelines

Use the clear ssh command to disconnect incoming or outgoing SSH connections. Incoming connections are managed by the SSH server running on the local networking device. Outgoing connections are initiated from the local networking device.

To display the session ID for a connection, use the show ssh command.

Examples

In the following example, the show ssh command is used to display all incoming and outgoing connections to the router. The clear ssh command is then used to terminate the incoming session with the ID number 0.

RP/0/RP0/CPU0:router# show ssh

SSH version: Cisco-2.0

session pty location state userid host ver

--------------------------------------------------------------------
Incoming sessions
0 vty0 0/33/1 SESSION_OPEN cisco 172.19.72.182 v2
1 vty1 0/33/1 SESSION_OPEN cisco 172.18.0.5 v2
2 vty2 0/33/1 SESSION_OPEN cisco 172.20.10.3 v1
3 vty3 0/33/1 SESSION_OPEN cisco 3333::50 v2

Outgoing sessions
1 0/33/1 SESSION_OPEN cisco 172.19.72.182 v2
2 0/33/1 SESSION_OPEN cisco 3333::50 v2
The following output is applicable for the `clear ssh` command starting release 6.0 and later.

```plaintext
RP/0/RP0/CPU0:router# clear ssh 0

RP/0/RP0/CPU0:router# show ssh

SSH version : Cisco-2.0

<table>
<thead>
<tr>
<th>id</th>
<th>chan</th>
<th>pty</th>
<th>location</th>
<th>state</th>
<th>userid</th>
<th>host</th>
<th>ver</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incoming sessions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>vty0</td>
<td>0/33/1</td>
<td>SESSION_OPEN</td>
<td>cisco</td>
<td>123.100.100.18</td>
<td>v2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Command-Line-Interface</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outgoing sessions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>0/33/1</td>
<td>SESSION_OPEN</td>
<td>cisco</td>
<td>172.19.72.182</td>
<td>v2</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>0/33/1</td>
<td>SESSION_OPEN</td>
<td>cisco</td>
<td>3333::50</td>
<td>v2</td>
</tr>
</tbody>
</table>

RP/0/RP0/CPU0:router# clear ssh 0
```
netconf-yang agent ssh

To enable netconf agent over SSH (Secure Shell), use the **netconf-yang agent ssh** command in the global configuration mode. To disable netconf, use the **no** form of the command.

```
netconf-yang agent ssh
no netconf-yang agent ssh
```

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

Global Configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

SSH is currently the supported transport method for Netconf.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>config-services</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to use the **netconf-yang agent ssh** command:

```
RP/0/RP0/CPU0:router (config) # netconf-yang agent ssh
```
To start the secure FTP (SFTP) client, use the sftp command.

```
sftp [username @ host : remote-filename] source-filename dest-filename [source-interface type interface-path-id] [vrf vrf-name]
```

**Syntax Description**

- **username** (Optional) Name of the user performing the file transfer. The at symbol (@) following the username is required.

- **hostname:remote-filename** (Optional) Name of the Secure Shell File Transfer Protocol (SFTP) server. The colon (:) following the hostname is required.

- **source-filename** SFTP source, including the path.

- **dest-filename** SFTP destination, including the path.

- **source-interface** (Optional) Specifies the source IP address of a selected interface for all outgoing SSH connections.

- **type** Interface type. For more information, use the question mark (?) online help function.

- **interface-path-id** Physical interface or virtual interface.

  **Note** Use the `show interfaces` command in XR EXEC mode to see a list of all interfaces currently configured on the router.

  For more information about the syntax for the router, use the question mark (?) online help function.

- **vrf vrf-name** Specifies the name of the VRF associated with the source interface.

**Command Default**

If no `username` argument is provided, the login name on the router is used. If no `hostname` argument is provided, the file is considered local.

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
</table>
| Release 6.0 | This command was introduced.

**Usage Guidelines**

SFTP provides for the secure (and authenticated) copying of files between a router and a remote host. Like the `copy` command, the `sftp` command can be invoked only in XR EXEC mode.

If a username is not provided, the login name on the router is used as the default. If a host name is not provided, the file is considered local.

If the source interface is specified in the `sftp` command, the `sftp` interface takes precedence over the interface specified in the `ssh client source-interface` command.
When the file destination is a local path, all of the source files should be on remote hosts, and vice versa. When multiple source files exist, the destination should be a preexisting directory. Otherwise, the destination can be either a directory name or destination filename. The file source cannot be a directory name.

If you download files from different remote hosts, that is, the source points to different remote hosts, the SFTP client spawns SSH instances for each host, which may result in multiple prompts for user authentication.

### Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>execute</td>
</tr>
<tr>
<td>basic-services</td>
<td>execute</td>
</tr>
</tbody>
</table>

### Examples

In the following example, user *abc* is downloading the file *ssh.diff* from the SFTP server *ena-view1* to *disk0*:

```
RP/0/RP0/CPU0:router#sftp abc@ena-view1:ssh.diff disk0
```

In the following example, user *abc* is uploading multiple files from *disk0:/sam_* to *users/abc/* on a remote SFTP server called *ena-view1*:

```
RP/0/RP0/CPU0:router# sftp disk0:/sam_* abc@ena-view1:/users/abc/
```

In the following example, user *admin* is downloading the file *run* from *disk0a:* to *disk0:/V6copy* on a local SFTP server using an IPv6 address:

```
RP/0/RP0/CPU0:router#sftp admin@[2:2:2::2]:disk0a:/run disk0:/V6copy
Connecting to 2:2:2::2...
Password:

disk0a:/run
   Transferred 308413 Bytes
   308413 bytes copied in 0 sec (338172)bytes/sec
RP/0/RP0/CPU0:router#dir disk0:/V6copy
Directory of disk0:
    70144 -rwx 308413 Sun Oct 16 23:06:52 2011 V6copy
2102657024 bytes total (1537638400 bytes free)
```

In the following example, user *admin* is uploading the file *v6copy* from *disk0:* to *disk0a:/v6back* on a local SFTP server using an IPv6 address:

```
RP/0/RP0/CPU0:router#sftp disk0:/V6copy admin@[2:2:2::2]:disk0a:/v6back
Connecting to 2:2:2::2...
Password:

disk0a:/V6copy
   Transferred 308413 Bytes
   308413 bytes copied in 0 sec (421329)bytes/sec
RP/0/RP0/CPU0:router#dir disk0a:/v6back
```
Directory of disk0a:

66016  -rwx  308413  Sun Oct 16 23:07:28 2011  v6back

2102788096 bytes total (2098987008 bytes free)

In the following example, user admin is downloading the file sampfile from disk0: to disk0a:/sampfile_v4 on a local SFTP server using an IPv4 address:

RP/0/RP0/CPU0:router#sftp admin@2.2.2.2:disk0:/sampfile disk0a:/sampfile_v4
Connecting to 2.2.2.2...
Password:

disk0:/sampfile
   Transferred 986 Bytes
   986 bytes copied in 0 sec (493000)bytes/sec

RP/0/RP0/CPU0:router#dir disk0a:/sampfile_v4

Directory of disk0a:

131520  -rwx  986  Tue Oct 18 05:37:00 2011  sampfile_v4

502710272 bytes total (502001664 bytes free)

In the following example, user admin is uploading the file sampfile_v4 from disk0a: to disk0:/sampfile_back on a local SFTP server using an IPv4 address:

RP/0/RP0/CPU0:router#sftp disk0a:/sampfile_v4 admin@2.2.2.2:disk0:/sampfile_back
Connecting to 2.2.2.2...
Password:

disk0a:/sampfile_v4
   Transferred 986 Bytes
   986 bytes copied in 0 sec (564000)bytes/sec

RP/0/RP0/CPU0:router#dir disk0:/sampfile_back

Directory of disk0:

121765  -rwx  986  Tue Oct 18 05:39:00 2011  sampfile_back

524501272 bytes total (512507614 bytes free)
**sftp (Interactive Mode)**

To enable users to start the secure FTP (SFTP) client, use the `sftp` command.

```
sftp [username @ host : remote-filename] [source-interface type interface-path-id] [vrf vrf-name]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>username</code></td>
<td>(Optional) Name of the user performing the file transfer. The at symbol (@) following the username is required.</td>
</tr>
<tr>
<td><code>hostname:remote-filename</code></td>
<td>(Optional) Name of the Secure Shell File Transfer Protocol (SFTP) server. The colon (:) following the hostname is required.</td>
</tr>
<tr>
<td><code>source-interface</code></td>
<td>(Optional) Specifies the source IP address of a selected interface for all outgoing SSH connections.</td>
</tr>
<tr>
<td><code>type</code></td>
<td>Interface type. For more information, use the question mark (?) online help function.</td>
</tr>
<tr>
<td><code>interface-path-id</code></td>
<td>Physical interface or virtual interface.</td>
</tr>
<tr>
<td><code>vrf vrf-name</code></td>
<td>Specifies the name of the VRF associated with the source interface.</td>
</tr>
</tbody>
</table>

**Command Default**

If no `username` argument is provided, the login name on the router is used. If no `hostname` argument is provided, the file is considered local.

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The SFTP client, in the interactive mode, creates a secure SSH channel where the user can enter any supported command. When a user starts the SFTP client in an interactive mode, the SFTP client process creates a secure SSH channel and opens an editor where user can enter any supported command.

More than one request can be sent to the SFTP server to execute the commands. While there is no limit on the number of ‘non-acknowledged’ or outstanding requests to the server, the server might buffer or queue these requests for convenience. Therefore, there might be a logical sequence to the order of requests.

The following unix based commands are supported in the interactive mode:

- `bye`
- `cd <path>`
The following commands are not supported:

- lcd, lls, lpwd, lumask, ln
- ln, symlink
- chgrp, chown
- !, !command
- ?
- mget, mput

In the following example, user admin is downloading and uploading a file from/to an external SFTP server using an IPv6 address:

```
RP/0/RP0/CPU0:router#sftp admin@[2::2]
Connecting to 2::2...
Password:
sftp> pwd
Remote working directory: /
sftp> cd /auto/tftp-server1-users5/admin
sftp> get frmRouter /disk0:/frmRouterdownload
/auto/tftp-server1-users5/admin/frmRouter
```
In the following example, user *abc* is downloading and uploading a file from/to an external SFTP server using an IPv4 address:

```
RP/0/RP0/CPU0:router#sftp abc@2.2.2.2
Connecting to 2.2.2.2...
Password:
sftp> pwd
Remote working directory: /
sftp> cd /auto/tftp-server1-users5/abc
sftp> get frmRouter /disk0:/frmRouterdownoad
/auto/tftp-server1-users5/abc/frmRouter
Transferred 1578 Bytes
1578 bytes copied in 0 sec (27684)bytes/sec
sftp> put /disk0:/frmRouterdownoad againtoServer
/disk0:/frmRouterdownoad
Transferred 1578 Bytes
1578 bytes copied in 0 sec (14747)bytes/sec
sftp>
```
show ssh

To display all incoming and outgoing connections to the router, use the `show ssh` command.

```
show  ssh
```

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `show ssh` command to display all incoming and outgoing Secure Shell (SSH) Version 1 (SSHv1) and SSH Version 2 (SSHv2) connections.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**

The following output is applicable for the `show ssh` command starting release 6.0 and later.

```
RP/0/RP0/CPU0:router# show ssh

SSH version : Cisco-2.0

id chan pty location state userid host ver
authentication connection type

Incoming sessions
0 1 vty0 0/33/1 SESSION_OPEN cisco 123.100.100.18 v2
password Command-Line-Interface

Outgoing sessions
1 0/33/1 SESSION_OPEN cisco 172.19.72.182 v2
2 0/33/1 SESSION_OPEN cisco 3333::50 v2

```

This table describes significant fields shown in the display.

**Table 10: show ssh Field Descriptions**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session</td>
<td>Session identifier for the incoming and outgoing SSH connections.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>chan</td>
<td>Channel identifier for incoming (v2) SSH connections. NULL for SSH v1 sessions.</td>
</tr>
<tr>
<td>pty</td>
<td>pty-id allocated for the incoming session. Null for outgoing SSH connection.</td>
</tr>
<tr>
<td>location</td>
<td>Specifies the location of the SSH server for an incoming connection. For an outgoing connection, location specifies from which route processor the SSH session is initiated.</td>
</tr>
<tr>
<td>state</td>
<td>The SSH state that the connection is currently in.</td>
</tr>
<tr>
<td>userid</td>
<td>Authentication, authorization and accounting (AAA) username used to connect to or from the router.</td>
</tr>
<tr>
<td>host</td>
<td>IP address of the remote peer.</td>
</tr>
<tr>
<td>ver</td>
<td>Specifies if the connection type is SSHv1 or SSHv2.</td>
</tr>
<tr>
<td>authentication</td>
<td>Specifies the type of authentication method chosen by the user.</td>
</tr>
<tr>
<td>connection type</td>
<td>Specifies which application is performed over this connection (Command-Line-Interface, Remote-Command, Scp, Sftp-Subsystem, or Netconf-Subsystem)</td>
</tr>
</tbody>
</table>
show ssh history

To display the last hundred SSH connections that were terminated, use the `show ssh history` command in XR EXEC mode.

**show ssh history**

**Syntax Description**
This command has no keywords or arguments.

**Command Default**
None

**Command Modes**
XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.4.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
No specific guidelines impact the use of this command.

**Examples**
The following is sample output from the `show ssh history` command to display the last hundred SSH sessions that were terminated:

```
RP/0/RP0/CP00:router# show ssh history

SSH version : Cisco-2.0

id | chan | pty | location | userid | host | ver | authentication
---|------|-----|----------|--------|------|-----|-----------------|
Incoming sessions
1  | 1    | XXXX| 0/RP0/CP00| root   | 10.105.227.252 | v2  | password
Netconf-Subsystem
2  | 1    | XXXX| 0/RP0/CP00| root   | 10.105.227.252 | v2  | password
Netconf-Subsystem
3  | 1    | XXXX| 0/RP0/CP00| root   | 10.105.227.252 | v2  | password
Netconf-Subsystem
4  | 1    | XXXX| 0/RP0/CP00| root   | 10.105.227.252 | v2  | password
Netconf-Subsystem
5  | 1    | XXXX| 0/RP0/CP00| root   | 10.105.227.252 | v2  | password
Netconf-Subsystem
6  | 1    | XXXX| 0/RP0/CP00| root   | 10.105.227.252 | v2  | password
Netconf-Subsystem
7  | 1    | XXXX| 0/RP0/CP00| root   | 10.105.227.252 | v2  | password
Netconf-Subsystem
8  | 1    | XXXX| 0/RP0/CP00| root   | 10.105.227.252 | v2  | password
```
Secure Shell Commands

show ssh history

Pty – VTY number used. This is represented as ‘XXXX’ when connection type is SFTP, SCP or Netconf.
show ssh history details

To display the last hundred SSH connections that were terminated, and also the start and end time of the session, use the `show ssh history details` command in XR EXEC mode.

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**

The following is sample output from the `show ssh history details` command to display the last hundred SSH sessions that were terminated along with the start and end time of the sessions:

```
RP/0/RP0/CPU0:router# show ssh history details

SSH version : Cisco-2.0

id      key-exchange      pubkey      incipher      outcipher      inmac
        start_time      end_time

Incoming Session
1 ecdh-sha2-nistp256  ssh-rsa  aes128-ctr  aes128-ctr  hmac-sha2-256
hmac-sha2-256  14-02-18 14:00:39  14-02-18 14:00:41
2 ecdh-sha2-nistp256  ssh-rsa  aes128-ctr  aes128-ctr  hmac-sha2-256
hmac-sha2-256  14-02-18 16:21:54  14-02-18 16:21:55
3 ecdh-sha2-nistp256  ssh-rsa  aes128-ctr  aes128-ctr  hmac-sha2-256
4 ecdh-sha2-nistp256  ssh-rsa  aes128-ctr  aes128-ctr  hmac-sha2-256
hmac-sha2-256  15-02-18 12:17:44  15-02-18 12:17:46
5 ecdh-sha2-nistp256  ssh-rsa  aes128-ctr  aes128-ctr  hmac-sha2-256
6 ecdh-sha2-nistp256  ssh-rsa  aes128-ctr  aes128-ctr  hmac-sha2-256
hmac-sha2-256  15-02-18 14:44:08  15-02-18 14:44:09
7 ecdh-sha2-nistp256  ssh-rsa  aes128-ctr  aes128-ctr  hmac-sha2-256
hmac-sha2-256  15-02-18 14:50:15  15-02-18 14:50:16
8 ecdh-sha2-nistp256  ssh-rsa  aes128-ctr  aes128-ctr  hmac-sha2-256
```
This table describes the significant fields shown in the display.

### Table 11: Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session</td>
<td>Session identifier for the incoming and outgoing SSH connections.</td>
</tr>
<tr>
<td>key-exchange</td>
<td>Key exchange algorithm chosen by both peers to authenticate each other.</td>
</tr>
<tr>
<td>pubkey</td>
<td>Public key algorithm chosen for key exchange.</td>
</tr>
<tr>
<td>incipher</td>
<td>Encryption cipher chosen for the receiver traffic.</td>
</tr>
<tr>
<td>outcipher</td>
<td>Encryption cipher chosen for the transmitter traffic.</td>
</tr>
<tr>
<td>inmac</td>
<td>Authentication (message digest) algorithm chosen for the receiver traffic.</td>
</tr>
<tr>
<td>outmac</td>
<td>Authentication (message digest) algorithm chosen for the transmitter traffic.</td>
</tr>
<tr>
<td>start_time</td>
<td>Start time of the session.</td>
</tr>
<tr>
<td>end_time</td>
<td>End time of the session.</td>
</tr>
</tbody>
</table>
show ssh session details

To display the details for all incoming and outgoing Secure Shell Version 2 (SSHv2) connections, use the show ssh session details command.

Syntax Description
This command has no keywords or arguments.

Command Default
None

Command Modes
XR EXEC mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines
Use the show ssh session details command to display a detailed report of the SSHv2 connections to or from the router, including the cipher chosen for the specific session.

Examples
The following is sample output from the show ssh session details command to display the details for all the incoming and outgoing SSHv2 connections:

```
RP/0/RP0/CPU0:router# show ssh session details

SSH version: Cisco-2.0
session key-exchange pubkey incipher outcipher inmac outmac
 ---------------------------------------------------------------
Incoming Session
0 diffie-hellman ssh-dss 3des-cbc 3des-cbc hmac-md5 hmac-md5
Outgoing connection
1 diffie-hellman ssh-dss 3des-cbc 3des-cbc hmac-md5 hmac-md5
```

This table describes the significant fields shown in the display.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>session</td>
<td>Session identifier for the incoming and outgoing SSH connections.</td>
</tr>
<tr>
<td>key-exchange</td>
<td>Key exchange algorithm chosen by both peers to authenticate each other.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>pubkey</td>
<td>Public key algorithm chosen for key exchange.</td>
</tr>
<tr>
<td>incipher</td>
<td>Encryption cipher chosen for the Rx traffic.</td>
</tr>
<tr>
<td>outcipher</td>
<td>Encryption cipher chosen for the Tx traffic.</td>
</tr>
<tr>
<td>inmac</td>
<td>Authentication (message digest) algorithm chosen for the Rx traffic.</td>
</tr>
<tr>
<td>outmac</td>
<td>Authentication (message digest) algorithm chosen for the Tx traffic.</td>
</tr>
</tbody>
</table>
To start the Secure Shell (SSH) client connection and enable an outbound connection to an SSH server, use the `ssh` command.

```plaintext
ssh [vrf vrf-name] {ipv4-address | ipv6-address | hostname} [username user-id] [cipher aes {128-cbc | 192-cbc | 256-cbc}] [source-interface type interface-path-id] [command command-name]
```

### Syntax Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vrf vrf-name</td>
<td>Specifies the name of the VRF associated with this connection.</td>
</tr>
<tr>
<td>ipv4-address</td>
<td>IPv4 address in A:B:C:D format.</td>
</tr>
<tr>
<td>ipv6-address</td>
<td>IPv6 address in X::X::X format.</td>
</tr>
<tr>
<td>hostname</td>
<td>Hostname of the remote node. If the hostname has both IPv4 and IPv6 addresses, the IPv6 address is used.</td>
</tr>
<tr>
<td>username user-id</td>
<td>(Optional) Specifies the username to use when logging in on the remote networking device running the SSH server. If no user ID is specified, the default is the current user ID.</td>
</tr>
<tr>
<td>cipher aes</td>
<td>(Optional) Specifies Advanced Encryption Standard (AES) as the cipher for the SSH client connection. <strong>Note</strong> If there is no specification of a particular cipher by the administrator, the client proposes 3DES as the default to ensure compatibility.</td>
</tr>
<tr>
<td>128-CBC</td>
<td>128-bit keys in CBC mode.</td>
</tr>
<tr>
<td>192-CBC</td>
<td>192-bit keys in CBC mode.</td>
</tr>
<tr>
<td>256-CBC</td>
<td>256-bit keys in CBC mode.</td>
</tr>
<tr>
<td>source interface</td>
<td>(Optional) Specifies the source IP address of a selected interface for all outgoing SSH connections.</td>
</tr>
<tr>
<td>type</td>
<td>Interface type. For more information, use the question mark (?) online help function.</td>
</tr>
<tr>
<td>interface-path-id</td>
<td>Physical interface or virtual interface. <strong>Note</strong> Use the <code>show interfaces</code> command in XR EXEC mode to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (?) online help function.</td>
</tr>
<tr>
<td>command</td>
<td>(Optional) Specifies a remote command. Adding this keyword prompts the SSHv2 server to parse and execute the <code>ssh</code> command in non-interactive mode instead of initiating the interactive session.</td>
</tr>
</tbody>
</table>

### Command Default

3DES cipher

### Command Modes

XR EXEC mode
# Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

## Usage Guidelines

Use the `ssh` command to make an outbound client connection. The SSH client tries to make an SSHv2 connection to the remote peer. If the remote peer supports only the SSHv1 server, it internally spawns an SSHv1 connection to the remote server. The process of the remote peer version detection and spawning the appropriate client connection is transparent to the user.

If a VRF is specified in the `ssh` command, the `ssh` interface takes precedence over the interface specified in the `ssh client source-interface`, on page 252 command.

When you configure the `cipher aes` keyword, an SSH client makes a proposal, including one or more of the key sizes you specified, as part of its request to the SSH server. The SSH server chooses the best possible cipher, based both on which ciphers that server supports and on the client proposal.

---

**Note**

AES encryption algorithm is not supported 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.

A VRF is required to run SSH, although this may be either the default VRF or a VRF specified by the user. If no VRF is specified while configuring the `ssh client source-interface`, on page 252 or `ssh client knownhost`, on page 251 commands, the default VRF is assumed.

Use the `command` keyword to enable the SSHv2 server to parse and execute the `ssh` command in non-interactive mode instead of initiating an interactive session.

## Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>execute</td>
</tr>
<tr>
<td>basic-services</td>
<td>execute</td>
</tr>
</tbody>
</table>

## Examples

The following sample output is from the `ssh` command to enable an outbound SSH client connection:

```
RP/0/RP0/CPU0:router# ssh vrf green username userabc
Password: 
Remote-host>
```
**ssh algorithms cipher**

To configure the list of supported SSH algorithms on the client or on the server, use the `ssh client algorithms cipher` command or `ssh server algorithms cipher` command in XR Config mode. To remove the configuration, use the `no` form of this command.

```plaintext
ssh {client | server} algorithms cipher {aes256-cbc | aes256-ctr | aes192-ctr | aes192-cbc | aes128-ctr | aes128-cbc | aes128-gcm@openssh.com | aes256-gcm@openssh.com | 3des-cbc}
```

**Syntax Description**

- **client**: Configures the list of supported SSH algorithms on the client.
- **server**: Configures the list of supported SSH algorithms on the server.

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

- **Task ID**: crypto
- **Operation**: read, write

This example shows how to enable CTR cipher on the client and CBC cipher on the server:

```plaintext
Router1#ssh client algorithms cipher aes128-ctr aes192-ctr aes256-ctr
Router1#ssh server algorithms cipher aes128-cbc aes192-cbc aes256-cbc 3des-cbc
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ssh client enable cipher</code>, on page 249</td>
<td>Enables CBC mode ciphers on the SSH client.</td>
</tr>
<tr>
<td><code>ssh server enable cipher</code>, on page 258</td>
<td>Enables CBC mode ciphers on the SSH server.</td>
</tr>
</tbody>
</table>
**ssh client enable cipher**

To enable the CBC mode ciphers 3DES-CBC and/or AES-CBC for an SSH client connection, use the `ssh client enable cipher` command in XR Config mode. To disable the ciphers, use the `no` form of this command.

```
ssh client enable cipher {aes-cbc | 3des-cbc}
```

**Syntax Description**

- `3des-cbc` Specifies that the 3DES-CBC cipher be enabled for the SSH client connection.
- `aes-cbc` Specifies that the AES-CBC cipher be enabled for the SSH client connection.

**Command Default**

CBC mode ciphers are disabled.

**Command Modes**

Global Configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.3.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The support for CBC ciphers were disabled by default, from Cisco IOS XR Software Release 6.1.2. Hence, `ssh client enable cipher` and `ssh server enable cipher` commands were introduced to explicitly enable CBC ciphers in required scenarios.

If a client tries to reach the router which acts as a server with CBC cipher, and if the CBC cipher is not explicitly enabled on that router, then the system displays an error message:

```
ssh root@x.x.x. -c aes128-cbc
Unable to negotiate with x.x.x.x port 22: no matching cipher found.
Their offer: aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com
```

You must configure `ssh server enable cipher aes-cbc` command in this case, to connect to the router using the CBC cipher.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to enable the 3DES-CBC and AES-CBC ciphers for an SSH client connection:

```
Router# configure
```
Router(config)# ssh client enable cipher aes-cbc 3des-cbc
Router(config)# commit

<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ssh algorithms cipher, on page 248</td>
<td>Configures the list of supported SSH algorithms on the client or on the server.</td>
</tr>
<tr>
<td></td>
<td>ssh server enable cipher, on page 258</td>
<td>Enables CBC mode ciphers on the SSH server.</td>
</tr>
</tbody>
</table>
**ssh client knownhost**

To authenticate a server public key (pubkey), use the `ssh client knownhost` command. To disable authentication of a server pubkey, use the `no` form of this command.

```plaintext
ssh client knownhost device:/filename
no ssh client knownhost device:/filename
```

**Syntax Description**

- `device:/filename`: Complete path of the filename (for example, slot0:/server_pubkey). The colon (:) and slash (/) are required.

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The server pubkey is a cryptographic system that uses two keys at the client end—a public key known to everyone and a private, or secret, key known only to the owner of the keys. In the absence of certificates, the server pubkey is transported to the client through an out-of-band secure channel. The client stores this pubkey in its local database and compares this key against the key supplied by the server during the early stage of key negotiation for a session-building handshake. If the key is not matched or no key is found in the local database of the client, users are prompted to either accept or reject the session.

The operative assumption is that the first time the server pubkey is retrieved through an out-of-band secure channel, it is stored in the local database. This process is identical to the current model adapted by Secure Shell (SSH) implementations in the UNIX environment.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following sample output is from the `ssh client knownhost` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh client knownhost disk0:/ssh.knownhost
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router# ssh host1 username user1234
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? yes
Password:
RP/0/RP0/CPU0:host1# exit
RP/0/RP0/CPU0:router# ssh host1 username user1234
```
### ssh client source-interface

To specify the source IP address of a selected interface for all outgoing Secure Shell (SSH) connections, use the `ssh client source-interface` command. To disable use of the specified interface IP address, use the `no` form of this command.

**Syntax Description**

<table>
<thead>
<tr>
<th>Command Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>type interface-path-id</code></td>
<td>Interface type. For more information, use the question mark (?) online help function.</td>
</tr>
<tr>
<td><code>no ssh client source-interface type interface-path-id</code></td>
<td>Physical interface or virtual interface.</td>
</tr>
</tbody>
</table>

**Note**

Use the `show interfaces` command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (?) online help function.

**Command Default**

No source interface is used.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `ssh client source-interface` command to set the IP address of the specified interface for all outgoing SSH connections. If this command is not configured, TCP chooses the source IP address when the socket is connected, based on the outgoing interface used—which in turn is based on the route required to reach the server. This command applies to outbound shell over SSH as well as Secure Shell File Transfer Protocol (SFTP) sessions, which use the ssh client as a transport.

The source-interface configuration affects connections only to the remote host in the same address family. The system database (Sysdb) verifies that the interface specified in the command has a corresponding IP address (in the same family) configured.

**Examples**

The following example shows how to set the IP address of the Management Ethernet interface for all outgoing SSH connections:
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh client source-interface MgmtEth 0/RP0/CPU0/0
ssh client vrf

To configure a new VRF for use by the SSH client, use the ssh client vrf command. To remove the specified VRF, use the no form of this command.

```
ssh client vrf vrf-name
no ssh client vrf vrf-name
```

**Syntax Description**

- `vrf-name` Specifies the name of the VRF to be used by the SSH client.

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

An SSH client can have only one VRF.

If a specific VRF is not configured for the SSH client, the default VRF is assumed when applying other SSH client-related commands, such as ssh client knownhost, on page 251 or ssh client source-interface, on page 252.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows the SSH client being configured to start with the specified VRF:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh client vrf green
```
**ssh server**

To bring up the Secure Shell (SSH) server, use the **ssh server** command. To stop the SSH server, use the **no** form of this command.

```
ssh server
no ssh server
```

This command has no keywords or arguments.

**Command Default**
The default SSH server version is 2 (SSHv2), which falls back to 1 (SSHv1) if the incoming SSH client connection is set to SSHv1.

**Command Modes**
XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
The SSH server listens for an incoming client connection on port 22. This server handles both Secure Shell Version 1 (SSHv1) and SSHv2 incoming client connections for both IPv4 and IPv6 address families. To accept only Secure Shell Version 2 connections, use the `ssh server v2, on page 262` command.

To verify that the SSH server is up and running, use the `show process sshd` command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

In the following example, how to bring up the SSH server:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh server
```
ssh server algorithms host-key

To configure the allowed SSH host-key pair algorithms from the list of auto-generated host-key pairs on the SSH server, use the `ssh server algorithms host-key` command in XR Config mode. To remove the configuration, use the `no` form of this command.

```
ssh server algorithms host-key {dsa | ecdsa-nistp256 | ecdsa-nistp384 | ecdsa-nistp521 | rsa}
```

Syntax Description:
- `dsa`
- `ecdsa-nistp256`
- `ecdsa-nistp384`
- `ecdsa-nistp521`
- `rsa`

Allows the specified DSA, ECDSA or RSA algorithm on the SSH server.
While configuring this, you can specify the algorithms in any order.

Command Default: None

Command Modes: XR Config mode

Command History:
```
Release  Modification
7.0.1      This command was introduced.
```

Usage Guidelines:
This configuration is optional. If this configuration is not present, it is considered that all the SSH host-key pairs are configured. In that case, the SSH client is allowed to connect to the SSH server with any of the host-key pairs.

You can also use the `crypto key zeroize` command to remove the SSH algorithms that are not required.

With the introduction of automatic generation of SSH host-key pairs, the `show crypto key mypubkey` command output displays the key-pairs for all the SSH host key algorithms. Prior to this, the command output displayed only the host-key pairs which were explicitly configured using the `crypto key generate` command.

Task ID:
```
Task ID  Operation
    crypto  read, write
```

This example shows how to select the `ecdsa` algorithm from the list of auto-generated host-key pairs on the SSH server:

```
Router# ssh server algorithms host-key ecdsa-nistp521
```
**ssh server disable hmac**

To disable HMAC cryptographic algorithm on the SSH server, use the `ssh server disable hmac` command, and to disable HMAC cryptographic algorithm on the SSH client, use the `ssh client disable hmac` command in XR Config mode. To disable this feature, use the `no` form of this command.

```
ssh {client | server} disable hmac {hmac-sha1 | hmac-sha2-512}
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>hmac-sha1</code></td>
<td>Disables the SHA-1 HMAC cryptographic algorithm.</td>
</tr>
<tr>
<td><code>hmac-sha2-512</code></td>
<td>Disables the SHA-2 HMAC cryptographic algorithm.</td>
</tr>
</tbody>
</table>

**Note** This option is available only for the `server`.  

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

This example shows how to disable SHA1 HMAC cryptographic algorithm on the SSH client:

```
Router#ssh client disable hmac hmac-sha1
```

This example shows how to disable SHA-2 HMAC cryptographic algorithm on the SSH server:

```
Router#ssh server disable hmac hmac-sha2-512
```
**ssh server enable cipher**

To enable CBC mode ciphers 3DES-CBC and/or AES-CBC for an SSH server connection, use the `ssh server enable cipher` command in XR Config mode. To disable the ciphers, use the `no` form of this command.

```
ssh server enable cipher {aes-cbc | 3des-cbc}
```

**Syntax Description**

- `3des-cbc` Specifies that the 3DES-CBC cipher be enabled for the SSH server connection.
- `aes-cbc` Specifies that the AES-CBC cipher be enabled for the SSH server connection.

**Command Default**

CBC mode ciphers are disabled.

**Command Modes**

Global Configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.3.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The support for CBC ciphers were disabled by default, from Cisco IOS XR Software Release 6.1.2. Hence, `ssh client enable cipher` and `ssh server enable cipher` commands were introduced to explicitly enable CBC ciphers in required scenarios.

**Task ID**

- **Task**: crypto
- **Operation**: read, write

**Examples**

The following example shows how to enable the 3DES-CBC and AES-CBC ciphers for an SSH server connection:

```
Router# configure
Router(config)# ssh server enable cipher aes-cbc 3des-cbc
Router(config)# commit
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ssh algorithms cipher</code>, on page 248</td>
<td>Configures the list of supported SSH algorithms on the client or on the server.</td>
</tr>
<tr>
<td><code>ssh client enable cipher</code>, on page 249</td>
<td>Enables CBC mode ciphers on the SSH client.</td>
</tr>
</tbody>
</table>
ssh server logging

To enable SSH server logging, use the ssh server logging command. To discontinue SSH server logging, use the no form of this command.

```
ssh server logging
no ssh server logging
```

Syntax Description

This command has no keywords or arguments.

Command Default

None

Command Modes

XR Config mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines

Only SSHv2 client connections are allowed.

Once you configure the logging, the following messages are displayed:

- Warning: The requested term-type is not supported
- SSH v2 connection from %s succeeded (user:%s, cipher:%s, mac:%s, pty:%s)

The warning message appears if you try to connect using an unsupported terminal type. Routers running the Cisco IOS XR software support only the vt100 terminal type.

The second message confirms a successful login.

Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

Examples

The following example shows the initiation of an SSH server logging:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh server logging
```
**ssh server rate-limit**

To limit the number of incoming Secure Shell (SSH) connection requests allowed per minute, use the `ssh server rate-limit` command. To return to the default value, use the `no` form of this command.

```
ssh server rate-limit rate-limit
no ssh server rate-limit
```

**Syntax Description**

- `rate-limit`: Number of incoming SSH connection requests allowed per minute. Range is from 1 to 120.

  - When setting it to 60 attempts per minute, it basically means that we can only allow 1 per second.
  - If you set up 2 sessions at the same time from 2 different consoles, one of them will get rate limited.
  - This is connection attempts to the ssh server, not bound per interface/username or anything like that. So value of 30 means 1 session per 2 seconds and so forth.

**Command Default**

- `rate-limit`: 60 connection requests per minute

**Command Modes**

- XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `ssh server rate-limit` command to limit the incoming SSH connection requests to the configured rate. Any connection request beyond the rate limit is rejected by the SSH server. Changing the rate limit does not affect established SSH sessions.

If, for example, the `rate-limit` argument is set to 30, then 30 requests are allowed per minute, or more precisely, a two-second interval between connections is enforced.

**Examples**

The following example shows how to set the limit of incoming SSH connection requests to 20 per minute:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh server rate-limit 20
```
ssh server session-limit

To configure the number of allowable concurrent incoming Secure Shell (SSH) sessions, use the ssh server session-limit command. To return to the default value, use the no form of this command.

**ssh server session-limit sessions**

**Syntax Description**

<table>
<thead>
<tr>
<th><strong>session</strong></th>
<th>Number of incoming SSH sessions allowed across the router. The range is from 1 to 100110.</th>
</tr>
</thead>
</table>

**Note**

Although CLI output option has 1024, you are recommended to configure session-limit not more than 100. High session count may cause resource exhaustion.

**Note**

From Cisco IOS XR release 6.4.1 and later, the session-limit is increased from 100 to 110.

**Command Default**

sessions: 64 per router

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th><strong>Release</strong></th>
<th><strong>Modification</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Release 6.4.1** The session-limit is increased from 100 to 110.

**Usage Guidelines**

Use the ssh server session-limit command to configure the limit of allowable concurrent incoming SSH connections. Outgoing connections are not part of the limit.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to set the limit of incoming SSH connections to 50:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh server session-limit 50
```
**ssh server v2**

To force the SSH server version to be only 2 (SSHv2), use the `ssh server v2` command. To bring down an SSH server for SSHv2, use the `no` form of this command.

```plaintext
ssh server v2
no ssh server v2
```

### Syntax Description
This command has no keywords or arguments.

### Command Default
None

### Command Modes
XR Config mode

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines
Only SSHv2 client connections are allowed.

### Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

### Examples
The following example shows how to initiate the SSH server version to be only SSHv2:

```plaintext
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh server v2
```
**ssh server vrf**

To bring up the Secure Shell (SSH) server and to configure one or more VRFs for its use, use the `ssh server vrf` command. To stop the SSH server from receiving any further connections for the specified VRF, use the `no` form of this command. Optionally ACLs for IPv4 and IPv6 can be used to restrict access to the server before the port is opened.

```
ssh server vrf  vrf-name  [ipv4 access-list  access-list name]  [ipv6 access-list  access-list name]
no ssh server vrf  vrf-name  [ipv4 access-list  access-list name]  [ipv6 access-list  access-list name]
```

**Syntax Description**
- **vrf vrf-name**
  - Specifies the name of the VRF to be used by the SSH server. The maximum VRF length is 32 characters.
  - **Note**: If no VRF is specified, the default VRF is assumed.
- **ipv4 access-list access-list name**
  - Configures an IPv4 access-list for access restrictions to the ssh server. The maximum length of the access-list name length is 32 characters.
- **ipv6 access-list access-list name**
  - Configures an IPv6 access-list for access restrictions to the ssh server. The maximum length of the access-list name length is 32 characters.

**Command Default**
The default SSH server version is 2 (SSHv2), which falls back to 1 (SSHv1) if the incoming SSH client connection is set to SSHv1.

**Command Modes**
XR Config mode

**Command History**
- **Release 6.0** This command was introduced.

**Usage Guidelines**
An SSH server must be configured at minimum for one VRF. If you delete all configured VRFs, including the default, the SSH server process stops. If you do not configure a specific VRF for the SSH client when applying other commands, such as `ssh client knownhost` or `ssh client source-interface` the default VRF is assumed.

To verify that the SSH server is up and running, use the `show process sshd` command.

**Examples**
In the following example, the SSH server is brought up to receive connections for VRF “green”:

```
RP/0/RP0/CPU0:router# configure
```
In the following example, the SSH server is brought up to receive connections for VRF “green” and a standard access list ipv4 access list named Internetfilter is configured:

```
RP/0/RP0/CPU0:router(config)# ssh server vrf green
```

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh server vrf green ipv4 access-list Internetfilter
```
**ssh server netconf**

To configure a port for the netconf SSH server, use the `ssh server netconf port` in the XR Config mode. To disable netconf for the configured port, use the `no` form of the command.

```
ssh server netconf [ port port-number ]
no ssh server netconf [ port port-number ]
```

### Syntax Description

`port-number` (Optional) Port number for the netconf SSH server (default port number is 830).

### Command Default

Default port number is 830.

### Command Modes

XR Config mode

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

No specific guidelines impact the use of this command.

### Task ID

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

### Example

This example shows how to use the `ssh server netconf port` command:

```
RP/0/RP0/CPU0:router (config) # ssh server netconf port 830
```
show tech-support ssh

To automatically run show commands that display system information, use the show tech-support command, use the `show tech-support ssh` command in XR EXEC mode.

`show tech-support ssh`

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.4.1</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read</td>
</tr>
</tbody>
</table>

**Examples**

The following is sample output from the `show tech-support ssh` command:

```
RP/0/RP0/CPU0:router# show tech-support ssh
++ Show tech start time: 2018-Feb-20.123016.IST ++
Tue Feb 20 12:30:27 IST 2018 Waiting for gathering to complete
............................
Tue Feb 20 12:32:35 IST 2018 Compressing show tech output
Show tech output available at 0/RP0/CPU0:/harddisk:/showtech/showtech-ssh-2018-Feb-20.123016.IST.tgz
++ Show tech end time: 2018-Feb-20.123236.IST ++
RP/0/RP0/CPU0:router#turin-secl#
```

The `show tech-support ssh` command collects the output of these CLI:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show logging</td>
<td>Displays the contents of the logging buffer.</td>
</tr>
<tr>
<td>show context location all</td>
<td>Displays the contents of the currently running configuration or a subset of that configuration.</td>
</tr>
<tr>
<td>show running-config</td>
<td>Displays the contents of the currently running configuration or a subset of that configuration.</td>
</tr>
<tr>
<td>show ip int brief</td>
<td>Displays brief information about each interface.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>show ssh</td>
<td>Displays all incoming and outgoing connections to the router.</td>
</tr>
<tr>
<td>show ssh session details</td>
<td>Displays the details for all the incoming and outgoing SSHv2 connections, to</td>
</tr>
<tr>
<td></td>
<td>the router.</td>
</tr>
<tr>
<td>show ssh rekey</td>
<td>Displays session rekey details such as session id, session rekey count, time</td>
</tr>
<tr>
<td></td>
<td>to rekey, data to rekey.</td>
</tr>
<tr>
<td>show ssh history</td>
<td>Displays the last hundred SSH connections that were terminated.</td>
</tr>
<tr>
<td>show tty trace info all all</td>
<td></td>
</tr>
<tr>
<td>show tty trace error all all</td>
<td></td>
</tr>
</tbody>
</table>
**ssh timeout**

To configure the timeout value for authentication, authorization, and accounting (AAA) user authentication, use the `ssh timeout` command. To set the timeout value to the default time, use the `no` form of this command.

```
ssh timeout seconds
no ssh timeout seconds
```

**Syntax Description**

`seconds` Time period (in seconds) for user authentication. The range is from 5 to 120.

**Command Default**

`seconds`: 30

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.0</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `ssh timeout` command to configure the timeout value for user authentication to AAA. If the user fails to authenticate itself within the configured time to AAA, the connection is terminated. If no value is configured, the default value of 30 seconds is used.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

In the following example, the timeout value for AAA user authentication is set to 60 seconds:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh timeout 60
```
Lawful Intercept Commands

Note

• Starting with Cisco IOS XR Release 6.6.25, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 560 Series Routers.

• Starting with Cisco IOS XR Release 6.3.2, all commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router.

• References to releases before Cisco IOS XR Release 6.3.2 apply to only the Cisco NCS 5500 Series Router.

• Cisco IOS XR Software Release 7.0.1 specific updates are not applicable for the following variants of Cisco NCS 540 Series Routers:
  • N540-28Z4C-SYS-A
  • N540-28Z4C-SYS-D
  • N540X-16Z4G8Q2C-A
  • N540X-16Z4G8Q2C-D
  • N540-12Z20G-SYS-A
  • N540-12Z20G-SYS-D
  • N540X-12Z16G-SYS-A
  • N540X-12Z16G-SYS-D

This module describes the commands used to configure Lawful intercept.

Note

All commands applicable for the Cisco NCS 5500 Series Router are also supported on the Cisco NCS 540 Series Router that is introduced from Cisco IOS XR Release 6.3.2. References to earlier releases in Command History tables apply to only the Cisco NCS 5500 Series Router.

• lawful-intercept disable, on page 270
lawful-intercept disable

To disable the Lawful Intercept (LI) feature, use the `lawful-intercept disable` command. To re-enable the LI feature, use the `no` form of this command.

```
lawful-intercept disable
no lawful-intercept disable
```

**Syntax Description**
This command has no keywords or arguments.

**Command Default**
LI feature is enabled by default only if the LI package is installed.

**Command Modes**
Global configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.1</td>
<td>This command is introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
If you disable lawful intercept, all Mediation Devices and associated TAPs are deleted.

To enable this command, you must install and activate the `ncs5500-li.rpm`.

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>li</td>
<td>read, write</td>
</tr>
</tbody>
</table>

This example shows how to configure the `lawful-intercept disable` command:

```
Router(config)# lawful-intercept disable
```
INDEX

C
conf-offset (macsec-policy) command  155
cryptographic-algorithm command  156

K
key (key chain) command  158, 159
key chain (key chain) command  160
key-server-priority (macsec-policy) command  163
key-string (keychain) command  117, 161

L
lawful-intercept disable command  270

M
macsec (key chain) command  166
macsec-policy command  168

S
security-policy command  170
send-lifetime command  164
show type 6 command  122

W
window-size command  171