THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

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)

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

This guide describes the configuration procedure and examples for system security in Cisco 8000 Series Routers.

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

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>March 2020</td>
<td>Initial release of this document.</td>
</tr>
<tr>
<td>August 2020</td>
<td>Republished for Release 7.1.2.</td>
</tr>
<tr>
<td>August 2020</td>
<td>Republished for Release 7.2.1.</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.

To use commands of this module, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using any command, contact your AAA administrator for assistance.

For detailed information about AAA concepts, configuration tasks, and examples, see the Configuring AAA Services chapter in the System Security Configuration Guide for Cisco 8000 Series Routers.

- aaa accounting, on page 3
- aaa accounting system default, on page 5
- aaa accounting system rp-failover, on page 7
- aaa accounting update, on page 8
- aaa authentication, on page 9
- aaa authorization, 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 (RADIUS), on page 29
- key (TACACS+), on page 31
- login authentication, on page 32
- password (AAA), on page 34
- radius-server dead-criteria time, on page 36
- radius-server dead-criteria tries, on page 38
- radius-server deadtime, on page 40
• radius-server host, on page 41
• radius-server key, on page 43
• radius-server retransmit, on page 44
• radius-server timeout, on page 45
• retransmit (RADIUS), on page 46
• secret, on page 47
• server (RADIUS), on page 49
• server (TACACS+), on page 51
• server-private (RADIUS), on page 52
• server-private (TACACS+), on page 54
• show aaa, on page 56
• show aaa password-policy, on page 62
• show radius accounting, on page 64
• show radius authentication, on page 66
• show radius, on page 68
• show radius dead-criteria, on page 70
• show radius server-groups, on page 72
• show tacacs, on page 74
• show tacacs server-groups, on page 76
• show user, on page 77
• single-connection, on page 81
• tacacs-server host, on page 82
• tacacs-server ipv4, on page 84
• tacacs-server key, on page 86
• tacacs-server timeout, on page 87
• tacacs source-interface, on page 88
• task, on page 90
• taskgroup, on page 91
• timeout login response, on page 93
• timeout (RADIUS), on page 94
• timeout (TACACS+), on page 95
• usergroup, on page 96
• username, on page 97
• users group, on page 101
• vrf (RADIUS), on page 103
• vrf (TACACS+), on page 104
**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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>commands</code></td>
<td>Enables accounting for XR EXEC shell commands.</td>
</tr>
<tr>
<td><code>exec</code></td>
<td>Enables accounting of a XR EXEC session.</td>
</tr>
<tr>
<td><code>mobile</code></td>
<td>Enables Mobile IP related accounting events.</td>
</tr>
<tr>
<td><code>network</code></td>
<td>Enables accounting for all network-related service requests, such as Internet Key Exchange (IKE) and Point-to-Point Protocol (PPP).</td>
</tr>
<tr>
<td><code>subscriber</code></td>
<td>Sets accounting lists for subscribers.</td>
</tr>
<tr>
<td><code>system</code></td>
<td>Enables accounting for all system-related events.</td>
</tr>
<tr>
<td><code>event manager</code></td>
<td>Sets the authorization list for XR EXEC.</td>
</tr>
<tr>
<td><code>default</code></td>
<td>Uses the listed accounting methods that follow this keyword as the default list of methods for accounting services.</td>
</tr>
<tr>
<td><code>list-name</code></td>
<td>Character string used to name the accounting method list.</td>
</tr>
<tr>
<td><code>start-stop</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>stop-only</code></td>
<td>Sends a “stop accounting” notice at the end of the requested user process.</td>
</tr>
<tr>
<td><code>none</code></td>
<td>Uses no accounting.</td>
</tr>
<tr>
<td><code>method</code></td>
<td>Method used to enable AAA system accounting. The value is one of the following options:</td>
</tr>
<tr>
<td></td>
<td>• <code>group tacacs+</code>—Uses the list of all TACACS+ servers for accounting.</td>
</tr>
<tr>
<td></td>
<td>• <code>group radius</code>—Uses the list of all RADIUS servers for accounting.</td>
</tr>
<tr>
<td></td>
<td>• <code>group named-group</code>—Uses a named subset of TACACS+ or RADIUS servers for accounting, as defined by the <code>aaa group server tacacs+</code> or <code>aaa group server radius</code> command.</td>
</tr>
</tbody>
</table>

### 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 7.0.12</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 {broadcast | 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.
- **broadcast**: Sets the broadcast 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 Config mode

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</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.
aaa accounting system default

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# aaa accounting system default start-stop group tacacs+
aaa accounting system rp-failover

To create an accounting list to send rp-failover or rp-switchover start or stop accounting messages, use the `aaa accounting system rp-failover` command in XR Config mode. To disable the system accounting for rp-failover, use the `no` form of this command.

```plaintext
aaa accounting system rp-failover {list_name | default} {start-stop | stop-only}
```

**Syntax Description**

- `list_name`: Specifies the accounting list name.
- `default`: Specifies the default accounting list.
- `start-stop`: Enables the start and stop records.
- `stop-only`: Enables the stop records only.

**Command Default**

None

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</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  Operation
aaa      read,
         write
```

This is an example of configuring the `aaa accounting system rp-failover` command for default accounting list:

```plaintext
RP/0/RP0/CPU0:router(config)# aaa accounting system rp-failover default start-stop none
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa attribute format</td>
<td>Create an AAA attribute format name.</td>
</tr>
</tbody>
</table>
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 7.0.12</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**

- `aaa` read,
- `write`

**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

To create a method list for authentication, use the `aaa authentication` command. To disable this authentication method, use the `no` form of this command.

```
aaa authentication {login | ppp} {default list-name | remote} method-list
no aaa authentication {login | onepk | ppp} {default list-name} method-list
```

### Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>login</code></td>
<td>Sets authentication lists for login.</td>
</tr>
<tr>
<td><code>onepk</code></td>
<td>Sets authentication lists for OnePk.</td>
</tr>
<tr>
<td><code>ppp</code></td>
<td>Sets authentication for Point-to-Point Protocol.</td>
</tr>
<tr>
<td><code>default</code></td>
<td>Uses the listed authentication methods that follow this keyword as the default list of methods for authentication.</td>
</tr>
<tr>
<td><code>list-name</code></td>
<td>Character string used to name the authentication method list.</td>
</tr>
<tr>
<td><code>method-list</code></td>
<td>Method used to enable AAA system accounting. The value is one of the following options:</td>
</tr>
<tr>
<td></td>
<td>• <code>group tacacs</code>—Specifies a method list that uses the list of all configured TACACS+ servers for authentication.</td>
</tr>
<tr>
<td></td>
<td>• <code>group radius</code>—Specifies a method list that uses the list of all configured RADIUS servers for authentication.</td>
</tr>
<tr>
<td></td>
<td>• <code>group named-group</code>—Specifies a method list that uses a named subset of TACACS+ or RADIUS servers for authentication, as defined by the <code>aaa group server tacacs+</code> or <code>aaa group server radius</code> command.</td>
</tr>
<tr>
<td></td>
<td>• <code>local</code>—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.</td>
</tr>
</tbody>
</table>

### Command Default

Default behavior applies the local authentication on all ports.

### Command Modes

Global configuration

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</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.

**Note**
- 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.

<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>

**Examples**

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

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

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa accounting, on page 3</td>
<td>Creates a method list for accounting.</td>
</tr>
<tr>
<td>aaa authorization, on page 11</td>
<td>Creates a method list for authorization.</td>
</tr>
<tr>
<td>aaa group server radius, on page 15</td>
<td>Groups different RADIUS server hosts into distinct lists and distinct methods.</td>
</tr>
<tr>
<td>aaa group server tacacs+, on page 17</td>
<td>Groups different TACACS+ server hosts into distinct lists and distinct methods.</td>
</tr>
<tr>
<td>login authentication, on page 32</td>
<td>Enables AAA authentication for logins.</td>
</tr>
<tr>
<td>tacacs-server host, on page 82</td>
<td>Specifies a TACACS+ host.</td>
</tr>
</tbody>
</table>
aaa authorization

To create a method list for authorization, use the **aaa authorization** command. To disable authorization for a function, use the **no** form of this command.

```
aaa authorization {commands | eventmanager | exec | network | subscriber | nacm} {default list-name}

no aaa authorization {commands | eventmanager | exec | network | subscriber | nacm}
```

### Syntax Description

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>commands</td>
<td>Configures authorization for all EXEC 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 (EXEC) 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>nacm</td>
<td>Enables the nacm functionality.</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 <strong>none</strong>, 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-name</td>
<td>Uses a named subset of TACACS+ or RADIUS servers for authorization as defined by the <strong>aaa group server tacacs+</strong> or <strong>aaa group server radius</strong> command.</td>
</tr>
</tbody>
</table>

### Command Default

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

### Command Modes

Global configuration

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>
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.

**Note**
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.

**Note**
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 EXEC mode commands a user issues. Command authorization attempts authorization for all EXEC mode commands.

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

- **EXEC authorization**—Applies authorization for starting an EXEC 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.

> Note 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.

### 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 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+
```

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa accounting, on page 3</td>
<td>Creates a method list for accounting.</td>
</tr>
</tbody>
</table>
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 7.0.12</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(config)# 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 7.0.12</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

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 7.0.12</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 an 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>7.0.12</td>
<td>This command was introduced.</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 8000 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 FIPS Mode chapter in the System Security Configuration Guide for Cisco 8000 Series Routers.

You must configure both authen-max-attempts and lockout-time in order for the lockout 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.

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>policy-name</td>
<td>None</td>
<td>253</td>
<td>1</td>
</tr>
<tr>
<td>max-length</td>
<td>253</td>
<td>253</td>
<td>2</td>
</tr>
<tr>
<td>min-length</td>
<td>2</td>
<td>253</td>
<td>2</td>
</tr>
<tr>
<td>special-char</td>
<td>0</td>
<td>253</td>
<td>0</td>
</tr>
<tr>
<td>upper-case</td>
<td>0</td>
<td>253</td>
<td>0</td>
</tr>
<tr>
<td>lower-case</td>
<td>0</td>
<td>253</td>
<td>0</td>
</tr>
<tr>
<td>numeric</td>
<td>0</td>
<td>253</td>
<td>0</td>
</tr>
<tr>
<td>For lifetime :</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>
<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
```
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 7.0.12</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

- **commands**: Enables authorization on the selected lines for all commands.
- **exec**: Enables authorization for an interactive XR EXEC mode session.
- **default**: Applies the default method list, created with the `aaa authorization` command.
- **eventmanager**: Sets eventmanager authorization method. This method is used for the embedded event manager.
- **list-name**: 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 `aaa authorization` command.

### 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 7.0.12</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 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 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 7.0.12</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: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
RP/0/RP0/CPU0:router(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 7.0.12</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.

```plaintext
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** 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 7.0.12</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 97 command in XR Config mode.

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.

**Syntax**

```
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**

- **Release 7.0.12**: This command was introduced.

**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 7.0.12</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 (RADIUS)

To specify the authentication and encryption key that is used between the router and the RADIUS daemon running on the RADIUS server, use the key (RADIUS) command in RADIUS server-group private configuration mode.

```
key {0 clear-text-key | 7 encrypted-key clear-text-key}
no key {0 clear-text-key | 7 encrypted-key clear-text-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) user password.

**Command Default**

For submode key commands, the default is to use the radius-server key command in global configuration mode, if defined. If the global key is also not defined, the configuration is not complete.

**Command Modes**

RADIUS server-group private configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</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>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

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

```
RP/0//CPU0:router# configure
RP/0//CPU0:router(config)# aaa group server radius group1
RP/0//CPU0:router(config-sg-radius)# server-private 10.1.1.1 auth-port 300
RP/0//CPU0:router(config-sg-radius-private)# key anykey
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa group server tacacs+, on page 17</td>
<td>Groups different RADIUS server hosts into distinct lists.</td>
</tr>
<tr>
<td>radius-server key, on page 43</td>
<td>Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>radius-server retransmit, on page 44</td>
<td>Specifies the number of times a RADIUS request is resent to a server if the server is not responding or is responding slowly.</td>
</tr>
<tr>
<td>server-private (RADIUS), on page 52</td>
<td>Configures the IP address of the private RADIUS server for the group server.</td>
</tr>
<tr>
<td>timeout (RADIUS), on page 94</td>
<td>Specifies the number of seconds the router waits for the RADIUS server to reply before retransmitting.</td>
</tr>
</tbody>
</table>
**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**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>0 clear-text-key</code></td>
<td>Specifies an unencrypted (cleartext) shared key.</td>
</tr>
<tr>
<td><code>7 encrypted-key</code></td>
<td>Specifies an encrypted shared key.</td>
</tr>
<tr>
<td><code>auth-key</code></td>
<td>Specifies the unencrypted key between the AAA server and the TACACS+ server.</td>
</tr>
</tbody>
</table>

**Command Default**

None

**Command Modes**

TACACS host configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</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
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tacacs-server host, on page 82</td>
<td>Specifies a TACACS+ host.</td>
</tr>
<tr>
<td>tacacs-server key, on page 86</td>
<td>Globally sets the authentication encryption key used for all TACACS+ communications between the router and the TACACS+ daemon.</td>
</tr>
</tbody>
</table>
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}
```

**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 7.0.12  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.

**Examples**

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

```
RP/0/RP0/CPU0:router# configure
```
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**

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

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</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 EXEC 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.
configure
username user1
password 0 pwd1
commit
show running-config
username user1
password 7 141B1309

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>group (AAA), on page 25</td>
<td>Adds a user to a group.</td>
</tr>
<tr>
<td>usergroup, on page 96</td>
<td>Accesses user group configuration mode and configures a user group, associating it with a set of task groups.</td>
</tr>
<tr>
<td>username, on page 97</td>
<td>Accesses username configuration mode and configures a new user with a username, establishing a password and granting permissions for that user.</td>
</tr>
<tr>
<td>line</td>
<td>Enters line template configuration mode for the specified line template. For more information, see the Cisco IOS XR System Management Command Reference.</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 global configuration 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.

### 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

Global configuration

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

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

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

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

### 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
```
<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>radius-server dead-criteria tries, on page 38</td>
<td>Specifies the number of consecutive timeouts that must occur on the router before the RADIUS server is marked as dead.</td>
</tr>
<tr>
<td></td>
<td>radius-server deadtime, on page 40</td>
<td>Defines the length of time, in minutes, for a RADIUS server to remain marked dead.</td>
</tr>
<tr>
<td></td>
<td>show radius dead-criteria, on page 70</td>
<td>Displays information for the dead-server detection criteria.</td>
</tr>
</tbody>
</table>
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. 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

Global configuration

Command History

Release | Modification
--- | ---
7.0.12 | This command was introduced.

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
```
<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>radius-server dead-criteria time, on page 36</td>
<td>Defines the length 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.</td>
</tr>
<tr>
<td></td>
<td>radius-server deadtime, on page 40</td>
<td>Defines the length of time, in minutes, for a RADIUS server to remain marked dead.</td>
</tr>
<tr>
<td></td>
<td>show radius dead-criteria, on page 70</td>
<td>Displays information for the dead-server detection criteria.</td>
</tr>
</tbody>
</table>
radius-server deadtime

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 XR Config mode. To set deadtime to 0, use the `no` form of this command.

```
radius-server  deadtime  minutes
```

**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. 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 7.0.12</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,</td>
</tr>
<tr>
<td></td>
<td>write</td>
</tr>
</tbody>
</table>

**Examples**

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

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

To specify a RADIUS server host, use the `radius-server host` command in XR Config mode. To delete the specified RADIUS host, use the `no` form of this command.

```
radius-server host 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.
  - IPv6 address is not supported.
- **auth-port port-number**: (Optional) Specifies the User Datagram Protocol (UDP) destination port for authentication requests; the host is not used for authentication if set to 0. If unspecified, the port number defaults to 1645.
- **acct-port port-number**: (Optional) Specifies the UDP destination port for accounting requests; the host is not used for accounting if set to 0. If unspecified, the port number defaults to 1646.
- **timeout seconds**: (Optional) The time interval (in seconds) that the router waits for the RADIUS server to reply before retransmitting. This setting overrides the global value of the `radius-server timeout` command. If no timeout value is specified, the global value is used. Enter a value in the range from 1 to 1000. Default is 5.
- **retransmit retries**: (Optional) The number of times a RADIUS request is re-sent to a server, if that server is not responding or is responding slowly. This setting overrides the global setting of the `radius-server retransmit` command. If no retransmit value is specified, the global value is used. Enter a value in the range from 1 to 100. Default is 3.
- **key string**: (Optional) Specifies the authentication and encryption key used between the router and 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.
  - The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the `radius-server host` command syntax. This is because the leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

### Command Default

No RADIUS host is specified; use global `radius-server` command values.

### Command Modes

XR Config mode

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

You can use multiple `radius-server host` commands to specify multiple hosts. The Cisco IOS XR software searches for hosts in the order in which you specify them.

If no host-specific timeout, retransmit, or key values are specified, the global values apply to each host.
<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operations ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa</td>
<td>read, write</td>
</tr>
</tbody>
</table>

### Examples

This example shows how to establish the host with IP address 172.29.39.46 as the RADIUS server, use ports 1612 and 1616 as the authorization and accounting ports, set the timeout value to 6, set the retransmit value to 5, and set “rad123” as the encryption key, matching the key on the RADIUS server:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# radius-server host 172.29.39.46 auth-port 1612 acct-port 1616
timeout 6 retransmit 5 key rad123
```

To use separate servers for accounting and authentication, use the zero port value as appropriate.

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>aaa accounting subscriber</code></td>
<td>Creates a method list for accounting.</td>
</tr>
<tr>
<td><code>aaa authentication subscriber</code></td>
<td>Creates a method list for authentication.</td>
</tr>
<tr>
<td><code>aaa authorization subscriber</code></td>
<td>Creates a method list for authorization.</td>
</tr>
<tr>
<td><code>radius-server key, on page 43</code></td>
<td>Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon.</td>
</tr>
<tr>
<td><code>radius-server retransmit, on page 44</code></td>
<td>Specifies how many times Cisco IOS XR software retransmits packets to a server before giving up.</td>
</tr>
<tr>
<td><code>radius-server timeout, on page 45</code></td>
<td>Sets the interval a router waits for a server host to reply.</td>
</tr>
</tbody>
</table>
radius-server key

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}
no radius-server key

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 clear-text-key</td>
<td>Specifies an unencrypted (cleartext) shared key.</td>
</tr>
<tr>
<td>7 encrypted-key clear-text-key</td>
<td>Specifies a encrypted shared key.</td>
</tr>
</tbody>
</table>

Command Default
The authentication and encryption key is disabled.

Command Modes
XR Config mode

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

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.

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 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

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 7.0.12</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**
- `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**

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**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**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**

<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 change the interval timer to 10 seconds:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# radius-server timeout 10
```
retransmit (RADIUS)

To specify the number of times a RADIUS request is resent to a server if the server is not responding or is responding slowly, use the `retransmit` command in RADIUS server-group private configuration mode.

```
retransmit retries
no retransmit retries
```

**Syntax Description**

`retries` 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.

**Command Default**

The default value is 3.

**Command Modes**

RADIUS server-group private configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Examples**

The following example shows how to set the retransmit value:

```
RP/0//CPU0:router# configure
RP/0//CPU0:router(config)# aaa group server radius group1
RP/0//CPU0:router(config-sg-radius)# server-private 10.1.1.1 auth-port 300
RP/0//CPU0:router(config-sg-radius-private)# retransmit 100
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaa group server tacacs+</td>
<td>Groups different RADIUS server hosts into distinct lists.</td>
</tr>
<tr>
<td>server-private (RADIUS)</td>
<td>Configures the IP address of the private RADIUS server for the group server.</td>
</tr>
<tr>
<td>timeout (RADIUS), on page 94</td>
<td>Specifies the number of seconds the router waits for the RADIUS server to reply before retransmitting.</td>
</tr>
</tbody>
</table>
**secret**

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</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. Otherwise, the password is not encrypted.</td>
</tr>
<tr>
<td>5</td>
<td>Specifies that an encrypted MD5 password (secret) follows.</td>
</tr>
<tr>
<td>8</td>
<td>(Optional) Specifies that SHA256-encrypted password follows.</td>
</tr>
<tr>
<td>9</td>
<td>(Optional) Specifies that scrypt-encrypted password follows.</td>
</tr>
<tr>
<td>10</td>
<td>(Optional) Specifies that SHA512-encrypted password follows.</td>
</tr>
</tbody>
</table>

*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 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

Type 10 encryption is applied as the default encryption type for the `secret` on Cisco IOS XR 64-bit operating systems.

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).
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.

<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 how to establish the clear-text secret “lab” for the user `user2`:

```
Router(config)# username user2
Router(config-un)# secret 0 lab
Router(config-un)# commit
```

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 97` command.

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

```
Router(config)# username user10 secret 10
$6$sPpIb7vy1d5wezT9g3kAFU33CLKj1/F.E4v/Hi.UnaqLwX9U6SeSBeZ9pgG6c5psbM3m2zg64j0l3q7mawxyu5VM/aBFJpge/jx2ZG17h6x9pWvJMK1
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(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**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**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.)
### 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 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 {hostnameip-address}
no server {hostnameip-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 7.0.12</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**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</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 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>Operations ID</th>
<th>aaa</th>
<th>read, write</th>
</tr>
</thead>
</table>

Examples

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

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

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

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

### Syntax Description

- **hostname**: Character string used to name the server host.
- **ip-address**: IP address of the TACACS+ server host. Both IPv4 and IPv6 addresses are supported.
- **port port-number**: (Optional) Specifies a server port number. This option overrides the default, which is port 49. Valid port numbers range from 1 to 65535.
- **timeout seconds**: (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 tacacs-server timeout command for only this server. The range is from 1 to 1000. The default is 5.
- **key string**: (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 tacacs-server key command. If no key string is specified, the global value is used.

### 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 7.0.12</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 tacacsp 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

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.

```
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 | askgroup-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**

EXEC
### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</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, 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-Key = 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
```
<table>
<thead>
<tr>
<th>Task</th>
<th>read</th>
<th>write</th>
<th>execute</th>
<th>debug</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bundle</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cdp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cef</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cgn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>config-mgmt</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>config-services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>crypto</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>diag</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>drivers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dwdm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eem</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eigrp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ethernet-services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ext-access</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fabric</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fault-mgr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>filesystem</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>firewall</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hdic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>host-services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hsrp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>inventory</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip-services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ipv4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ipv6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>isis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>l2vpn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>li</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>logging</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lpts</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>monitor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mpls-ldp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mpls-static</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mpls-te</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>multicast</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>netflow</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>network</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ospf</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ouni</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pkg-mgmt</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pos-dpt</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ppp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qos</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rib</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rip</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>root-lr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>route-map</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>route-policy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sbc</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>snmp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sonet-advb</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>syslog</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>system</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>transport</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tty-access</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tunnel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>universal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vlan</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vrrp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

System Security Command Reference for Cisco 8000 Series Routers
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:

```
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 root system. The task-group root system has the following combined set of task IDs, which includes all inherited groups:

```
Task: aaa : READ  WRITE  EXECUTE  DEBUG
Task: aaa acl : READ  WRITE  EXECUTE  DEBUG
Task: admin atm : READ  WRITE  EXECUTE  DEBUG
Task: atm basic-services : READ  WRITE  EXECUTE  DEBUG
Task: basic-services bcdl : READ  WRITE  EXECUTE  DEBUG
Task: bcdl bfd : READ  WRITE  EXECUTE  DEBUG
Task: bfd bgp : READ  WRITE  EXECUTE  DEBUG
Task: bgp boot : READ  WRITE  EXECUTE  DEBUG
Task: boot bundle : READ  WRITE  EXECUTE  DEBUG
Task: bundle cdp : READ  WRITE  EXECUTE  DEBUG
Task: cdp cef : READ  WRITE  EXECUTE  DEBUG
Task: cef config-mgmt : READ  WRITE  EXECUTE  DEBUG
Task: config-mgmt services : READ  WRITE  EXECUTE  DEBUG
Task: crypto : READ  WRITE  EXECUTE  DEBUG
Task: diag drivers : READ  WRITE  EXECUTE  DEBUG
Task: ext-access fabric : READ  WRITE  EXECUTE  DEBUG
Task: fault-mgr filesystem : READ  WRITE  EXECUTE  DEBUG
Task: filesystem fr : READ  WRITE  EXECUTE  DEBUG
Task: fr hdic : READ  WRITE  EXECUTE  DEBUG
Task: hdic host-services : READ  WRITE  EXECUTE  DEBUG
Task: host-services harp : READ  WRITE  EXECUTE  DEBUG
Task: hsrp interface : READ  WRITE  EXECUTE  DEBUG
Task: interface inventory : READ  WRITE  EXECUTE  DEBUG
Task: inventory ip-services : READ  WRITE  EXECUTE  DEBUG
Task: ipv4 : READ  WRITE  EXECUTE  DEBUG
Task: ipv4 ipv6 : READ  WRITE  EXECUTE  DEBUG
Task: ipv6 isis : READ  WRITE  EXECUTE  DEBUG
Task: isis logging : READ  WRITE  EXECUTE  DEBUG
Task: logging lpts : READ  WRITE  EXECUTE  DEBUG
Task: lpts monitor : READ  WRITE  EXECUTE  DEBUG
Task: mpls-ldp static : READ  WRITE  EXECUTE  DEBUG
Task: mpls-static te : READ  WRITE  EXECUTE  DEBUG
Task: mpls-te multicast : READ  WRITE  EXECUTE  DEBUG
Task: multicast netflow : READ  WRITE  EXECUTE  DEBUG
Task: netflow network : READ  WRITE  EXECUTE  DEBUG
Task: ospf : READ  WRITE  EXECUTE  DEBUG
Task: ospf ouni : READ  WRITE  EXECUTE  DEBUG
Task: ouni pkg-mgmt : READ  WRITE  EXECUTE  DEBUG
Task: pkg pos-mgmt dpt : 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
Task: root-system : READ  WRITE  EXECUTE  DEBUG
Task: route-map : READ  WRITE  EXECUTE  DEBUG
Task: route-policy : 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: transport : READ  WRITE  EXECUTE  DEBUG
Task: tty-access : READ  WRITE  EXECUTE  DEBUG
Task: universal : READ  WRITE  EXECUTE  DEBUG
Task: vlan : READ  WRITE  EXECUTE  DEBUG
Task: vrrp : READ  WRITE  EXECUTE  DEBUG

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

```
Username lab
User group root-lr
User group cisco-support
```

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
bodl
bfd
bqp
boot
bundle
cdp
cef
cisco-support
config-mgmt
config-services
crypto
diag
disallowed
drivers
eigrp
ext-access
fabric
fault-mgr
filesystem
firewall
fr
hdlc
host-services
hsrp
interface
inventory
ip-services
ipv4
```
Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show user, on page 77</td>
<td>Displays task IDs enabled for the currently logged-in user.</td>
</tr>
</tbody>
</table>
### 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]
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Policy Name</th>
<th>Specifies the name of password policy.</th>
</tr>
</thead>
</table>

- **Command Default**: None
- **Command Modes**: XR EXEC mode
- **Command History**:
  - **Release**: 7.0.12
  - **Modification**: This command was introduced.

- **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 out 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
```
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 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 7.0.12</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>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 2: 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.

**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 7.0.12</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>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 3: 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

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 7.0.12</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>aaa</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 4: 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 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**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host ip-addr</td>
<td>Specifies the name or IP address of the configured RADIUS server.</td>
</tr>
<tr>
<td>auth-port auth-port</td>
<td>(Optional) Specifies the authentication port for the RADIUS server. The default value is 1645.</td>
</tr>
<tr>
<td>acct-port acct-port</td>
<td>(Optional) Specifies the accounting port for the RADIUS server. The default value is 1646.</td>
</tr>
</tbody>
</table>

**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 7.0.12</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>aaa</td>
<td>read</td>
</tr>
</tbody>
</table>

**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 5: show radius dead-criteria 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>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**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>group-name</td>
<td>(Optional) Name of the server group. The properties are displayed.</td>
</tr>
<tr>
<td>detail</td>
<td>(Optional) Displays properties for all the server groups.</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 7.0.12</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 deadlock value is displayed and this message is omitted. The following sample output is for the show radius server-groups command:

```plaintext
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 'radgrpl' 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>
<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.

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 7.0.12</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: 1.2.3.5/49 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>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Server IP address.</td>
</tr>
<tr>
<td>open</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.

**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>7.0.12</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 8: show tacacs server-groups Field Descriptions**

<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 7.0.12</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.

**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: 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: hdc : 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: multicast : READ WRITE EXECUTE DEBUG
Task: netflow : READ WRITE EXECUTE DEBUG
Task: network : READ WRITE EXECUTE DEBUG
Task: ospf : READ WRITE EXECUTE DEBUG
Task: oui : 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: 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: hrp : 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>sysmgr</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>
**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 7.0.12</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.

```plaintext
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 tacacs-server timeout command for this server only. The valid timeout range is from 1 to 1000 seconds. Default is 5. Note: You can use this parameter only in the config-tacacs-host sub-mode.</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. Note: You can use this parameter only in the config-tacacs-host sub-mode.</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 7.0.12</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
RP/0/RP0/CPU0:router(config-tacacs-host)#
```

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 ipv4
tacacs-server ipv4 dscp dscp-value

Syntax Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv4</td>
<td>Specifies the dscp bit for the IPv4 packets.</td>
</tr>
<tr>
<td>dscp</td>
<td>Sets the DSCP in the IP header.</td>
</tr>
</tbody>
</table>

dscp-value Specifies the options for setting the value of DSCP. The available options are:

- <0-63> Differentiated services codepoint value
- af11 Match packets with AF11 dscp (001010)
- af12 Match packets with AF12 dscp (001100)
- af13 Match packets with AF13 dscp (001110)
- af21 Match packets with AF21 dscp (010010)
- af22 Match packets with AF22 dscp (010100)
- af23 Match packets with AF23 dscp (010110)
- af31 Match packets with AF31 dscp (011010)
- af32 Match packets with AF32 dscp (011100)
- af33 Match packets with AF33 dscp (011110)
- af41 Match packets with AF41 dscp (100010)
- af42 Match packets with AF42 dscp (100100)
- af43 Match packets with AF43 dscp (100110)
- cs1 Match packets with CS1(precedence 1) dscp (001000)
- cs2 Match packets with CS2(precedence 2) dscp (010000)
- cs3 Match packets with CS3(precedence 3) dscp (011000)
- cs4 Match packets with CS4(precedence 4) dscp (100000)
- cs5 Match packets with CS5(precedence 5) dscp (101000)
- cs6 Match packets with CS6(precedence 6) dscp (110000)
- cs7 Match packets with CS7(precedence 7) dscp (111000)
- default Match packets with default dscp (000000)
- ef Match packets with EF dscp (101110)
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 7.0.12</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, 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-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 auth-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 7.0.12</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.

**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 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 7.0.12</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**

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

**Examples**

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

```
RP/0/RP0/CPU0:router(config)# tacacs-server timeout 10
```
**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**

- `type` Interface type. For more information, use the question mark (?) online help function.
- `path-id` Physical interface or virtual interface.

**Note** Use the `show interfaces` command in XR Config 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-id` 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 7.0.12</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
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**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>Enables read-only privileges for the named task ID.</td>
</tr>
<tr>
<td>write</td>
<td>Enables write privileges for the named task ID. The term “write” implies read also.</td>
</tr>
<tr>
<td>execute</td>
<td>Enables execute privileges for the named task ID.</td>
</tr>
<tr>
<td>debug</td>
<td>Enables debug privileges for the named task ID.</td>
</tr>
<tr>
<td><code>taskid-name</code></td>
<td>Name of the task ID.</td>
</tr>
</tbody>
</table>

**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 7.0.12</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.

**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
```
# taskgroup

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}]
nostaskgroup taskgroup-name
```

## Syntax Description

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

## 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 7.0.12</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 taskgroup 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.
### 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 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 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 7.0.12</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/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# line template alpha
RP/0/RP0/CPU0:router(config-line)# timeout login response 20
```
timeout (RADIUS)

To specify the number of seconds the router waits for the RADIUS server to reply before retransmitting, use the `timeout` command in RADIUS server-group private 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**

- RADIUS server-group private configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</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`, `write`

**Examples**

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

```
RP/0//CPU0:router# configure
RP/0//CPU0:router(config)# aaa group server radius group1
RP/0//CPU0:router(config-sg-radius)# server-private 10.1.1.1 auth-port 300
RP/0//CPU0:router(config-sg-radius-private)# timeout 500
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>radius-server timeout</code>, on page 45</td>
<td>Sets the interval for which a router waits for a server host to reply before timing out.</td>
</tr>
<tr>
<td><code>radius-server retransmit</code>, on page 44</td>
<td>Specifies the number of times a RADIUS request is resent to a server if the server is not responding or is responding slowly.</td>
</tr>
<tr>
<td><code>server-private (RADIUS)</code>, on page 52</td>
<td>Configures the IP address of the private RADIUS server for the group server.</td>
</tr>
</tbody>
</table>
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 7.0.12</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
```
**usergroup**

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.

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

**Syntax Description**

- `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 7.0.12</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` 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:

```plaintext
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. To delete a user from the database, use the `no` form of this command.

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

**Syntax Description**

- **user-name**
  - Name of the user. The `user-name` 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.

- **password**
  - (Optional) Enables a password to be created for the named user.

- **0**
  - (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. Can be up to 253 characters in length.

- **7**
  - (Optional) Specifies that an encrypted password follows.

- **password**
  - (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.

- **password-policy**
  - Applies a particular password policy to the user.

- **policy-name**
  - Specifies the name of the password policy. This policy name has to be configured prior to applying this policy to the user.
<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 usergroup command.</td>
</tr>
</tbody>
</table>

**Command Default**

No usernames are defined in the system.

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

- A user is never allowed to have cisco-support privileges as the only group.
- Type 10 (SHA512) is the default password type for the secret configuration.

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 **aaa authentication** command.

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

**Note**

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.

For more details on defining a password policy, refer **aaa password-policy** command. The AAA password security policy feature works as such for Cisco IOS XR platforms. Whereas, it is supported only on XR VM, for Cisco IOS XR 64 bit platforms.

### Examples

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

```plaintext
Router# config
Router(config)# username user1
Router(config-un)# ?
```

<table>
<thead>
<tr>
<th>clear</th>
<th>Clear the uncommitted configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>commit</td>
<td>Commit the configuration changes to running</td>
</tr>
<tr>
<td>describe</td>
<td>Describe a command without taking real actions</td>
</tr>
<tr>
<td>do</td>
<td>Run an exec command</td>
</tr>
<tr>
<td>exit</td>
<td>Exit from this submode</td>
</tr>
<tr>
<td>group</td>
<td>User group in which this user will be a member of</td>
</tr>
<tr>
<td>no</td>
<td>Negate a command or set its defaults</td>
</tr>
<tr>
<td>password</td>
<td>Specify the password for the user</td>
</tr>
<tr>
<td>pwd</td>
<td>Commands used to reach current submode</td>
</tr>
<tr>
<td>root</td>
<td>Exit to the XR Config mode</td>
</tr>
<tr>
<td>secret</td>
<td>Specify the secure password for the user</td>
</tr>
</tbody>
</table>
show | Show contents of configuration

Router(config-un)#

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

Router# configure
Router(config)# username user1
Router(config-un)# password 0 password1

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 47 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
$8$ZYKGl1dZIw73D1$i1UWJOqTLoMyExhsNKoL5vMtvCOYguM5ajXf4uGeQj6I
Router(config-un)# commit

This example shows how to configure Type 9 password:

Router# configure
Router(config)# username user9 secret 9
$9$r1QLB3rp1RLB$5o2fLWKFYH6B/kApxkkXmIqbPAHrRZkPEoh3WqGbwwQ
Router(config-un)# commit

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

Router# configure
Router(config)# username user10 secret 10
$6$9UvJidvsTEg9bAFP3L1EI/I.F.EAv/Hi.UseGwX80sSEr9ApG6c5pzhM3m2tbgM4jObAqQ7meAwyhu5VM/aR9Jge/jx2ZG17h6xPrvJMF1
Router(config-un)# 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}
```

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

**Syntax Description**

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

- **cisco-support**
  
  Specifies that users logging in through the line are given Cisco support personnel privileges.

- **maintenance**
  
  Specifies that users logging in through the line are given SCAPA maintenance privileges.

- **netadmin**
  
  Specifies that users logging in through the line are given network administrator privileges.

- **operator**
  
  Specifies that users logging in through the line are given operator privileges.

- **provisioning**
  
  Specifies that users logging in through the line are given SCAPA provisioning privileges.

- **retrieve**
  
  Specifies that users logging in through the line are given SCAPA retrieve privileges.

- **root-lr**
  
  Specifies that users logging in through the line are given root logical router (LR) privileges.

- **serviceadmin**
  
  Specifies that users logging in through the line are given service administrator group privileges.

- **sysadmin**
  
  Specifies that users logging in through the line are given system administrator privileges.
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 7.0.12</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 7.0.12</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.

**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 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 7.0.12</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**
- `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
```
IPSec Commands

This module describes the IPSec commands.

For detailed information about the configuration tasks, and examples, see the *System Security Configuration Guide for Cisco 8000 Series Routers*.

- clear crypto ipsec sa, on page 106
- interface tunnel-ip (GRE), on page 107
- show crypto ipsec sa, on page 108
- show crypto ipsec summary, on page 111
- show crypto ipsec transform-set, on page 113
clear crypto ipsec sa

To delete specific security associations (SAs), or all SAs in the IP Security (IPSec) security associations database (SADB), use the **clear crypto ipsec sa** command.

```
clear crypto ipsec sa {sa-id | all | counters | {sa-id | all} | interface tunnel-ipsec}
```

### Syntax Description

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sa-id</code></td>
<td>Identifier for the SA. IPSec supports from 1 to 64,500 sessions.</td>
</tr>
<tr>
<td><code>all</code></td>
<td>Deletes all IPSec SAs in the IPSec SADB.</td>
</tr>
<tr>
<td><code>counters</code></td>
<td>Clears the counters in the IPSec SADB.</td>
</tr>
<tr>
<td><code>interface</code></td>
<td>Clears the interfaces in the IPSec SADB.</td>
</tr>
<tr>
<td><code>tunnel-ipsec</code></td>
<td>The range of tunnel-ipsec is &lt;0-4294967295&gt;.</td>
</tr>
</tbody>
</table>

### Command Default

No default behavior or values

### Command Modes

EXEC

### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

SAs are established to secure data flows in IPSec. Use the **clear crypto ipsec sa** command to delete active IPSec sessions or force IPSec to reestablish new SAs. Usually, the establishment of SAs is negotiated between peers through Internet Key Exchange (IKE) on behalf of IPSec.

### Task ID

<table>
<thead>
<tr>
<th>Task</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>execute</td>
</tr>
</tbody>
</table>

### Examples

The following example shows how to remove the SA with ID 100 from the SADB:

```
RP/0/RP0/CPU0:router# clear crypto ipsec sa 100
```

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show crypto ipsec sa, on page 108</td>
<td>Displays the settings used by current SAs.</td>
</tr>
</tbody>
</table>
interface tunnel-ip (GRE)

To configure a tunnel interface for generic routing encapsulation (GRE), use the `interface tunnel-ip` command in global configuration mode. To delete the IP tunnel interface, use the `no` form of this command.

```
interface tunnel-ip number
no interface tunnel-ip number
```

**Syntax Description**

- `number` Instance number of the interface. The range is from 0 to 65535.

**Command Default**

None

**Command Modes**

Global configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</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>interface</td>
<td>read, write</td>
</tr>
</tbody>
</table>

**Examples**

The following example shows how to use the `interface tunnel-ip` command:

```
RP/0//CPU0:router# configure
RP/0//CPU0:router(config)# interface tunnel-ip 50000
RP/0//CPU0:router(config-if)#
```
show crypto ipsec sa

To display security association (SA) information based on the rack/slot/module location, use the show crypto ipsec sa command.

```
show crypto ipsec sa [ { sa-id | peer ip-address | profile profile-name | detail | count | fvrf fvrf-name | ivrf ivrf-name | location node-id } ]
```

**Syntax Description**

- **sa-id**: (Optional) Identifier for the SA. The range is from 1 to 64500.
- **peer ip-address**: (Optional) IP address used on the remote (PC) side. Invalid IP addresses are not accepted.
- **profile profile-name**: (Optional) Specifies the alphanumeric name for a security profile. The character range is from 1 to 64. Profile names cannot be duplicated.
- **detail**: (Optional) Provides additional dynamic SA information.
- **count**: (Optional) Provides SA count.
- **fvrf fvrf-name**: (Optional) Specifies that all existing SAs for front door virtual routing and forwarding (FVRF) is the same as the fvrf-name.
- **ivrf ivrf-name**: (Optional) Specifies that all existing SAs for inside virtual routing and forwarding (IVRF) is the same as the ivrf-name.
- **location node-id**: (Optional) Specifies that the SAs are configured on a specified location.

**Command Modes**

- EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

If no optional argument or keyword is used, all SAs are displayed within a flow. Within a flow, the SAs are listed by protocol (Encapsulating Security Payload [ESP] or Authentication Header [AH]) and direction (inbound or outbound).

The **detail** keyword provides additional information only for SAs that are configured in a software crypto engine. The SAs are configured by using tunnel-ipsec and transport.

**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 sample output is from the **show crypto ipsec sa** command:

```
RP/0/RP0/CPU0:router# show crypto ipsec sa
```
SSA id: 510
Node id: 0/1/0
SA Type: MANUAL
interface: service-ipsec22
profile: p7
local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.255/512/0)
remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/512/0)
local crypto endpt: 0.0.0.0, remote crypto endpt: 0.0.0.0, vrf default

#pkts tx :0 #pkts rx :0
#bytes tx :0 #bytes rx :0
#pkts encrypt :0 #pkts decrypt :0
#pkts digest :0 #pkts verify :0
#pkts encrypt fail:0 #pkts decrypt fail:0
#pkts digest fail:0 #pkts verify fail:0
#pkts replay fail:0
#pkts tx errors :0 #pkts rx errors :0

outbound esp sas:
  spi: 0x322(802)
  transform: esp-3des-md5
  in use settings = Tunnel
  sa agreed lifetime: 3600s, 4194303kb
  sa timing: remaining key lifetime: 3142303931sec/0kb
  sa DPD: disable, mode none, timeout 0s
  sa idle timeout: disable, 0s
  sa anti-replay (HW accel): enable, window 64

inbound esp sas:
  spi: 0x322(802)
  transform: esp-3des-md5
  in use settings = Tunnel
  sa agreed lifetime: 3600s, 4194303kb
  sa timing: remaining key lifetime: 3142303931sec/0kb
  sa DPD: disable, mode none, timeout 0s
  sa idle timeout: disable, 0s
  sa anti-replay (HW accel): enable, window 64

This table describes the significant fields shown in the display.

Table 9: show crypto ipsec sa Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA id</td>
<td>Identifier for the SA.</td>
</tr>
<tr>
<td>interface</td>
<td>Identifier for the interface.</td>
</tr>
<tr>
<td>profile</td>
<td>String of alphanumeric characters that specify the name of a security profile.</td>
</tr>
<tr>
<td>local ident</td>
<td>IP address, mask, protocol, and port of the local peer.</td>
</tr>
<tr>
<td>remote ident</td>
<td>IP address, mask, protocol and port of the remote peer.</td>
</tr>
<tr>
<td>outbound esp sas</td>
<td>Outbound ESP SAs.</td>
</tr>
<tr>
<td>inbound esp sas</td>
<td>Inbound ESP SAs.</td>
</tr>
<tr>
<td>transform</td>
<td>The transform being used in the SA.</td>
</tr>
</tbody>
</table>
The following sample output is from the `show crypto ipsec sa` command for the `profile` keyword for a profile named pn1:

```
RP/0/RP0/CPU0:router# show crypto ipsec sa profile pn1

SA id: 2
interface: tunnel0
profile: pn1
local ident (addr/mask/prot/port): (172.19.70.92/255.255.255.255/0/0)
remote ident (addr/mask/prot/port): (172.19.72.120/255.255.255.255/0/0)
local crypto endpt: 172.19.70.92, remote crypto endpt: 172.19.72.120
outbound esp sas:
spi: 0x8b0e950f (2332988687)
transform: esp-3des-sha
in use settings = Tunnel
sa lifetime: 3600s, 4194303kb

SA id: 2
interface: tunnel0
profile: pn1
local ident (addr/mask/prot/port): (172.19.72.120/255.255.255.255/0/0)
remote ident (addr/mask/prot/port): (172.19.70.92/255.255.255.255/0/0)
local crypto endpt: 172.19.72.120, remote crypto endpt: 172.19.70.92
inbound esp sas:
spi: 0x2777997c (662149500)
transform: esp-3des-sha
in use settings = Tunnel
sa lifetime: 3600s, 4194303kb
```

The following sample output is from the `show crypto ipsec sa` command for the `peer` keyword:

```
RP/0/RP0/CPU0:router# show crypto ipsec sa peer 172.19.72.120

SA id: 2
interface: tunnel0
profile: pn1
local ident (addr/mask/prot/port): (172.19.70.92/255.255.255.255/0/0)
remote ident (addr/mask/prot/port): (172.19.72.120/255.255.255.255/0/0)
local crypto endpt: 172.19.70.92, remote crypto endpt: 172.19.72.120
outbound esp sas:
spi: 0x8b0e950f (2332988687)
transform: esp-3des-sha
in use settings = Tunnel
sa lifetime: 3600s, 4194303kb

SA id: 2
interface: tunnel0
profile: pn1
local ident (addr/mask/prot/port): (172.19.72.120/255.255.255.255/0/0)
remote ident (addr/mask/prot/port): (172.19.70.92/255.255.255.255/0/0)
local crypto endpt: 172.19.72.120, remote crypto endpt: 172.19.70.92
inbound esp sas:
spi: 0x2777997c (662149500)
transform: esp-3des-sha
in use settings = Tunnel
sa lifetime: 3600s, 4194303kb
```
show crypto ipsec summary

To display IP Security (IPSec) summary information, use the **show crypto ipsec summary** command.

```bash
show crypto ipsec summary
```

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7.2</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 ipsec summary** command:

```
RP/0/RP0/CPU0:router# show crypto ipsec summary
# * Attached to a transform indicates a bundle
# Active IPSec Sessions: 1
SA Interface Local Peer/Port Remote Peer/Port FVRF Profile Transform Lifetime
------------------------------------------------------------------------------------------
502 tunnel-ipsec100 70.70.70.2/500 60.60.60.2/500 default ipsec1 esp-3des esp 3600/10000000
```

This table describes the significant fields shown in the display.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>Identifier for the security association.</td>
</tr>
<tr>
<td>Node</td>
<td>Identifier for the node.</td>
</tr>
<tr>
<td>Local Peer</td>
<td>IP address of the local peer.</td>
</tr>
<tr>
<td>Remote Peer</td>
<td>IP address of the remote peer.</td>
</tr>
</tbody>
</table>

---

System Security Command Reference for Cisco 8000 Series Routers
### show crypto ipsec summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FVRF</td>
<td>The front door virtual routing and forwarding (FVRF) of the SA. If the FVRF is global, the output shows f_vrf as an empty field</td>
</tr>
<tr>
<td>Mode</td>
<td>Profile mode type.</td>
</tr>
<tr>
<td>Profile</td>
<td>Crypto profile in use.</td>
</tr>
<tr>
<td>Transform</td>
<td>Transform in use.</td>
</tr>
<tr>
<td>Lifetime</td>
<td>Lifetime value, displayed in seconds followed by kilobytes.</td>
</tr>
</tbody>
</table>
show crypto ipsec transform-set

To display the configured transform sets, use the `show crypto ipsec transform-set` command.

```
show crypto ipsec transform-set [transform-set-name]
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th><code>transform-set-name</code> (Optional) IPSec transform set with the specified value for the <code>transform-set-name</code> argument are displayed.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Command Default</th>
<th>No default values. The default behavior is to print all the available transform-sets.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Command Modes</th>
<th>EXEC</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Command History</th>
<th>Release 7.0.12  This command was introduced.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Usage Guidelines</th>
<th>If no transform is specified, all transforms are displayed.</th>
</tr>
</thead>
</table>

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

<table>
<thead>
<tr>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>The following sample output is from the <code>show crypto ipsec transform-set</code> command:</td>
</tr>
</tbody>
</table>

```
RP/0/RP0/CPU0:router# show crypto ipsec transform-set

Transform set combined-des-sha: {esp-des esp-sha-hmac}
Transform set tsfm2: {esp-md5-hmac esp-3des }
    Mode: Transport
Transform set tsfm1: {esp-md5-hmac esp-3des }
    Mode: Tunnel
Transform set ts1: {esp-des }
    Mode: Tunnel
```
show crypto ipsec transform-set
Keychain Management Commands

This module describes the commands used to configure keychain management.

For detailed information about keychain management concepts, configuration tasks, and examples, see the Implementing Keychain Management on configuration module in the System Security Configuration Guide for Cisco 8000 Series Routers.

- accept-lifetime, on page 116
- accept-tolerance, on page 117
- cryptographic-algorithm, on page 118
- key chain (key chain), on page 120
- key (key chain), on page 121
- key-string (keychain), on page 122
- send-lifetime, on page 124
- show key chain, on page 125
**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**

- **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 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. The range is from 1-2147483646.
- **infinite** (Optional) Specifies that the key never expires after 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**

None

**Command Modes**

Key configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</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, 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 7.0.12</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**
- `system` read, write

**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
```
**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>Syntax</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>7.0.12</td>
<td>This command was introduced.</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.
Keychain Management Commands

<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, 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 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>7.0.12</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 (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**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**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**

<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(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 (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
```

**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.
  - Encrypted key strings—Minimum of 4 characters and no maximum.

**Command Default**
The default value is clear.

**Command Modes**
Keychain-key configuration

**Command History**
- **Release 7.0.12** This command was introduced.

**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**
- **50ae fd**

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

**Examples**
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.

```
send-lifetime start-time [{duration duration value | infinite end-time}]
no send-lifetime start-time [{duration duration value | 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**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</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, write</td>
</tr>
</tbody>
</table>

**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 7.0.12</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 key chain
MACsec Encryption Commands

This module describes the commands used to configure MACsec encryption.

For detailed information about MACsec concepts, configuration tasks, and examples, see the Configuring MACsec chapter in the System Security Configuration Guide for Cisco 8000 Series Routers.

- cipher-suite, on page 128
- conf-offset, on page 129
- hw-module macsec-fips-post, on page 130
- hw-module macsec-mode, on page 132
- key, on page 134
- key-server-priority, on page 135
- key chain, on page 136
- key-string, on page 137
- lifetime, on page 139
- macsec-policy, on page 141
- macsec shutdown, on page 144
- show hw-module macsec-fips-post, on page 145
- show hw-module macsec-mode, on page 147
- window-size, on page 149
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 remove this configuration, 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>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**

- system  read, write

**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)# commit
```
conf-offset

Confines the confidentiality offset for MACsec encryption in the MACsec policy configuration mode. To remove this configuration, 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 bytes.
  - CONF-OFFSET-50: Offsets the encryption by 50 bytes.

**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>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**

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

**Examples**

The following example shows how to use the conf-offset command:

```sh
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)#
```
**hw-module macsec-fips-post**

To enable the power-on self-test (POST) known answer test (KAT) for the physical layer transceiver (PHY) of a line card, use the `hw-module macsec-fips-post` command in XR Config mode mode. To remove this configuration, use the no form of this command.

```
hw-module macsec-fips-post location { location | all }
```

**Syntax Description**
- **location**: Enables POST KAT for a specific node location.
- **all**: Enables POST KAT for all nodes.

**Command Default**
Disabled by default

**Command Modes**
XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.14</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**
You must reload the line card for this configuration to take effect.

You can use the `show hw-module macsec-fips-post` command to know the current mode of POST KAT configuration, and what action is to be performed.

**Note**
If power-on self-test (POST) known answer test (KAT) is already enabled on the PHY, then the system does not allow you to configure the `hw-module macsec-fips-post location all` command again. This restriction is in place to prevent conflicts in configuration, especially in a configuration restore scenario. In such scenarios, you can make use of the `show hw-module macsec-mode fips-post` command to know of the respective running configurations in place.

**Task ID**

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

This example shows how to enable power-on self-test KAT for the physical layer transceiver (PHY) of a line card:

```
Router# configure
Router(config)# hw-module macsec-fips-post location 0/4/CPU0
```
Router(config)# commit

<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show hw-module macsec-fips-post, on page 145</td>
<td>Displays the power-on self-test (POST) known answer test (KAT) configurations of nodes in a router.</td>
</tr>
</tbody>
</table>
### hw-module macsec-mode

To enable the MACsec mode for the physical layer transceiver (PHY) of a line card, use the `hw-module macsec-mode` command in XR Config mode mode. To remove this configuration, use the no form of this command.

#### Syntax

```
hw-module macsec-mode location {all | location}
```

#### Syntax Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>Specifies the node location to enable the MACsec mode.</td>
</tr>
<tr>
<td>all</td>
<td>Enables MACsec mode for all nodes.</td>
</tr>
<tr>
<td>location</td>
<td>Enables MACsec mode for a specific node.</td>
</tr>
</tbody>
</table>

#### Command Default

Disabled by default

#### Command Modes

XR Config mode

#### Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

#### Usage Guidelines

This configuration helps to avoid interface flap when MACsec is configured on an interface.

You must reload the line card for this configuration to take effect.

You can use the `show hw-module macsec-mode` command to know the current mode of MACsec, and what action is to be performed.

### Note

If the MACsec mode is already enabled on a node such as a line card, then the system does not allow you to configure the `hw-module macsec-mode location all` command again. This restriction is in place to prevent conflicts in configuration, especially in a configuration restore scenario. In such scenarios, you can make use of the `show hw-module macsec-mode` command to know of the respective running configurations in place.

#### Task ID

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

This example shows how to enable the MACsec mode for the physical layer transceiver (PHY) of a line card:

```
Router# configure
Router(config)# hw-module macsec-mode location 0/1/CPU0
```
Router(config)# commit

<table>
<thead>
<tr>
<th>Related Commands</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show hw-module macsec-mode, on page 147</td>
<td>Displays the MACsec mode of a line card and the user action to be performed.</td>
<td></td>
</tr>
</tbody>
</table>
key

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

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

**Syntax Description**

- `key-id`  Hexadecimal string of 2-64 characters.

**Command Default**

No default behavior or values.

**Command Modes**

Key chain configuration

**Command History**

- **Release** 7.0.12  This command was introduced.

**Usage Guidelines**

The key must be of even number of hex characters. Entering an odd number of characters will exit the MACsec configuration mode.

**Task ID**

- **Task**  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# key chain mac_chain macsec
RP/0/RP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
```
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 remove this configuration, 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>7.0.12</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)#
```
key chain

To create or modify a key chain, use the key chain command in the key chain configuration mode. To remove this configuration, use the no form of this command.

```
key chain key-chain-name macsec
```

**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.
- `macsec`: Specifies the key chain for MACsec encryption.

**Command Modes**

Key chain configuration

**Command Default**

No default behavior or values

**Command History**

- **Release 7.0.12**: This command was introduced.

**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 key configuration submode under the macsec key chain mode.

To remove this configuration, use the **no** form of this command.

```plaintext
key-string  [clear | password | password6]  key-string-text  cryptographic-algorithm  {aes-128-cmac | aes-256-cmac}
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>clear</strong></td>
<td></td>
<td>Specifies the key string in clear-text form.</td>
</tr>
<tr>
<td><strong>password</strong></td>
<td></td>
<td>Specifies the key in encrypted form.</td>
</tr>
<tr>
<td><strong>password6</strong></td>
<td></td>
<td>Specifies the key in Type 6 encrypted form.</td>
</tr>
<tr>
<td><strong>key-string-text</strong></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 (128-bit encryption)/64 (256-bit encryption) characters (hexadecimal string).</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**

Key configuration submode under the macsec key chain mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</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

**Task ID**

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

System Security Command Reference for Cisco 8000 Series Routers
The following examples show how to use the `key-string` 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
12345678123456781234567812345678 cryptographic-algorithm AES-128-CMAC
```

' For AES 256-bit encryption with clear-text CAK:

```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 clear
12345678123456781234567812345678123456781234567812345678123456781234567812345678123456781234567812345678 cryptographic-algorithm AES-256-CMAC
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# commit
```
lifetime

Configures the validity period for the MACsec key or CKN in the Keychain-key configuration mode. To remove this configuration, 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 2020 to Dec 31 2020), 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

| start-time | Start time in hh:mm:ss from which the key becomes valid. The range is from 0:0:0 to 23:59:59. |
| end-time | End time in hh:mm:ss at which point the key becomes invalid. The range is from 0:0:0 to 23:59:59. |
| start_date | The date in DD month YYYY format when the key becomes valid. |
| end_date | The date in DD month YYYY format when the key becomes invalid. |
| duration validity | The key chain is valid for the duration you configure. You can configure duration in seconds. The range is from 1 to 2147483646. |
| infinite | The key chain is valid indefinitely. |

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>7.0.12</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 lifetime command: 

! For AES 128-bit encryption
lifetime

For AES 256-bit encryption, with lifetime specified as duration:

```plaintext
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
123456781234567812345678123456781234567812345678123456781234567812345678
cryptographic-algorithm AES-256-CMAC
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# lifetime 05:00:00 20 february 2020 duration 2592000
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# commit
```

Lifetime specified as infinite:

```plaintext
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
12345678123456781234567812345678123456781234567812345678123456781234567812345678123456781234567812345678123456781234567812345678123456781234567812345678
cryptographic-algorithm AES-256-CMAC
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# lifetime 05:00:00 20 february 2020 infinite
RP/0/RP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# commit
```
macsec-policy

Creates a MACsec policy for MACsec encryption in the global configuration mode. To remove this configuration, use the no form of this command.


Syntax Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy_name</td>
<td>The MACsec policy name with a maximum length of 16.</td>
</tr>
<tr>
<td>cipher-suite</td>
<td>Specifies the cipher-suite used for encryption.</td>
</tr>
<tr>
<td>conf-offset</td>
<td>Specifies the confidentiality offset value for encryption.</td>
</tr>
<tr>
<td>delay-protection</td>
<td>Enables data delay protection.</td>
</tr>
<tr>
<td>include-icv-indicator</td>
<td>Includes integrity check value (ICV) indicator parameter set in MACsec Key Agreement PDU (MKPDU).</td>
</tr>
<tr>
<td>key-server-priority</td>
<td>Specifies the key-server priority for the node.</td>
</tr>
<tr>
<td>policy-exception</td>
<td>Specifies MACsec policy exception to allow packets in clear text.</td>
</tr>
<tr>
<td>lacp-in-clear</td>
<td>Allows Link Aggregation Control Plane protocol (LACP) packets in clear text.</td>
</tr>
<tr>
<td>sak-rekey-interval</td>
<td>Specifies the interval after which the key-server generates a new Secure Association Key (SAK) for a secured session.</td>
</tr>
<tr>
<td>security-policy</td>
<td>Specifies the security policy as must secure or should secure for data encryption.</td>
</tr>
<tr>
<td>use-eapol-pae-in-icv</td>
<td>Enables the use of Extensible Authentication Protocol over LAN (EAPoL) port access entity (POE) address in ICV.</td>
</tr>
<tr>
<td>vlan-tags-in-clear</td>
<td>Specifies the number of vlan-tags in clear (1 or 2).</td>
</tr>
<tr>
<td>window-size</td>
<td>Specifies the window-size used for encryption.</td>
</tr>
<tr>
<td></td>
<td>The range is from 0 to 1024.</td>
</tr>
</tbody>
</table>

Command Default

No default behavior or values.

Command Modes

Global configuration

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>
This example shows how to configure 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)#
```

This example shows how to specify the cipher-suite used for MACsec encryption:

```
RP/0/RP0/CPU0:router(config-mac_policy)# cipher-suite GCM-AES-128
RP/0/RP0/CPU0:router(config-mac_policy)#
```

This example shows how to specify the confidentiality offset value used for MACsec encryption:

```
RP/0/RP0/CPU0:router(config-mac_policy)# conf-offset CONF-OFFSET-30
RP/0/RP0/CPU0:router(config-mac_policy)#
```

This example shows how to enable data delay protection under the macsec policy:

```
RP/0/RP0/CPU0:router(config-mac_policy)# delay-protection
RP/0/RP0/CPU0:router(config-mac_policy)#
```

This example shows how to include ICV indicator under the macsec policy:

```
RP/0/RP0/CPU0:router(config-mac_policy)# include-icv-indicator
RP/0/RP0/CPU0:router(config-mac_policy)#
```

This example shows how to specify the key-server priority for the node:

```
RP/0/RP0/CPU0:router(config-mac_policy)# key-server-priority 10
RP/0/RP0/CPU0:router(config-mac_policy)#
```

This example shows how to specify the macsec policy exception to allow packets in clear text:

```
RP/0/RP0/CPU0:router(config-mac_policy)# policy-exception lacp-in-clear
RP/0/RP0/CPU0:router(config-mac_policy)#
```

This example shows how to specify the SAK rekey interval under the macsec policy:

```
RP/0/RP0/CPU0:router(config-mac_policy)# sak-rekey-interval seconds 86400
RP/0/RP0/CPU0:router(config-mac_policy)#
```

This example shows how to specify the security policy as must-secure or should-secure under the macsec policy:
MACsec Encryption Commands

This example shows how to enable the use of EAPoL PAE address in ICV:

```
RP/0/RP0/CPU0:router(config-mac_policy)# security-policy must-secure
RP/0/RP0/CPU0:router(config-mac_policy)#
```

```
RP/0/RP0/CPU0:router(config-mac_policy)# use-eapol-pae-in-icv
RP/0/RP0/CPU0:router(config-mac_policy)#
```

This example shows how to specify the number of vlan-tags in clear:

```
RP/0/RP0/CPU0:router(config-mac_policy)# vlan-tags-in-clear 1
RP/0/RP0/CPU0:router(config-mac_policy)#
```

This example shows how to specify the window-size under the macsec-policy:

```
RP/0/RP0/CPU0:router(config-mac_policy)# window-size 256
RP/0/RP0/CPU0:router(config-mac_policy)#
```
macsec shutdown

To enable MACsec shutdown, use the `macsec shutdown` command. 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

Global configuration

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</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

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

Example

The following example shows how to enable MACsec shutdown:

```
RP/0/RSP0/CP00:router# configure terminal
RP/0/RSP0/CP00:router(config)# macsec shutdown
```
**show hw-module macsec-fips-post**

To display the power-on self-test (POST) known answer test (KAT) configurations of nodes in a router, use the `show hw-module macsec-mode` command in the XR EXEC mode.

```
show hw-module macsec-fips-post [location { location | all }]
```

**Syntax Description**

- `location` Displays the POST KAT configuration for a node location.
- `location` Specifies the node location for which the POST KAT configuration is to be displayed.
- `all` Displays the POST KAT configuration for all nodes.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.14</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>system</td>
<td>read</td>
</tr>
</tbody>
</table>

This example shows how to view the POST KAT configuration of all nodes in a router:

Before location reload:

```
Router#show hw-module macsec-fips-post location all
Wed Jun 17 09:36:31.932 UTC

Location Configured Applied Action
---------------------------------------------
0/0/CPU0 NO NO NONE
0/11/CPU0 YES NO RELOAD
```

After location reload:

```
Router#show hw-module macsec-fips-post location all
Wed Jun 17 10:03:57.263 UTC

Location Configured Applied Action
---------------------------------------------
0/0/CPU0 NO NO NONE
```
### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>hw-module macsec-fips-post, on page 130</code></td>
<td>Enables power-on self-test known answer test (KAT) for the physical layer transceiver (PHY) of a line card</td>
</tr>
</tbody>
</table>
**show hw-module macsec-mode**

To display the MACsec mode of line cards, and the user action to be performed, use the `show hw-module macsec-mode` command in the XR EXEC mode.

```
show hw-module macsec-mode [ location { location | all } ]
```

**Syntax Description**

- `location location`: Specifies the location of the line card for which the MACsec mode and the user action to be performed are to be displayed.
- `all`: Displays the MACsec mode information for all the nodes.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
<tr>
<td>7.0.14</td>
<td>This command was modified to include the <code>all</code> option.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

- `system read`

This example shows how to view the MACsec mode of all nodes and the user action to be performed:

```
Router# show hw-module macsec-mode
Sun Feb 16 21:06:07.726 UTC
Location Configured Running Action
-------------------------------------------------------------
0/0/CPU0 NO NO NONE
0/7/CPU0 YES YES NONE
```

You can also use the `show hw-module macsec-mode location all` command to display the MACsec mode information of all nodes. This `location all` option is available starting Cisco IOS XR Software Release 7.0.14.

This example shows how to view the MACsec mode of a specific node and the user action to be performed:

```
Router# show hw-module macsec-mode location 0/1/CPU0
Sat Dec 7 14:31:52.668 UTC
Location Configured Running Action
0/1/CPU0 YES YES NONE
```
After performing the specified action (reload, in this case):

Router#show hw-module macsec-mode location 0/1/CPU0
Sat Dec 7 15:01:00.463 UTC

<table>
<thead>
<tr>
<th>Location</th>
<th>Configured</th>
<th>Running</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/1/CPU0</td>
<td>YES</td>
<td>YES</td>
<td>NONE</td>
</tr>
</tbody>
</table>

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hw-module macsec-mode, on page 132</td>
<td>Enables the MACsec mode for the physical layer transceiver (PHY) of a line card.</td>
</tr>
</tbody>
</table>
window-size

Configures the replay protection window size in MACsec policy configuration mode. To remove this configuration, 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`

**Syntax Description**

| value | Number of out-of-sequence frames that can be accepted at the interface without being dropped. The range is 0-1024. |

**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>7.0.12</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
```
window-size
Management Plane Protection Commands

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

For detailed information about management plane protection concepts, configuration tasks, and examples, see the Implementing Management Plane Protection chapter in the System Security Configuration Guide for Cisco 8000 Series Routers.

- address ipv4 (MPP), on page 152
- address ipv6 (MPP), on page 153
- allow, on page 154
- control-plane, on page 156
- inband, on page 157
- interface (MPP), on page 158
- management-plane, on page 160
- out-of-band, on page 161
- show mgmt-plane, on page 162
- vrf (MPP), on page 164
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 7.0.12</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, 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-telnetftp-peer)# 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**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>peer-ip-address</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>peer-ip-address/length</code></td>
<td>Prefix of the peer IPv6 address.</td>
</tr>
</tbody>
</table>

**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 7.0.12</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>ID</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>read, write</td>
</tr>
</tbody>
</table>

**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-inband)# interface HundredGigabitEthernet 0/1/1/2
RP/0/RP0/CPU0:router(config-mpp-inband-if)# 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 7.0.12</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**

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

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
```
To enter the control plane configuration mode, use the `control-plane` command. To disable all the configurations under control plane mode, use the `no` form of this command.

```
control-plane
no control-plane
```

### 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>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines
Use the `control-plane` command to enter control plane 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 enter control plane configuration mode using the `control-plane` command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# control-plane
RP/0/RP0/CPU0:router(config-ctrl)#
```
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 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `inband` command to enter management plane protection inband 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 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**
- **Release 7.0.12** This command was introduced.

**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.

**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
```
The following example shows how to configure all out-of-band interfaces for MPP:

```
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)#
```
management-plane

To configure management plane protection to allow and disallow protocols, use the `management-plane` command in control plane configuration mode. To disable all configurations under management-plane mode, use the `no` form of this command.

```
management-plane
no management-plane
```

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

Control plane configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Use the `management-plane` command to enter the management plane protection configuration mode.

**Examples**

The following example shows how to enter management plane protection configuration mode using the `management-plane` 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

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.

**out-of-band**

**no out-of-band**

**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 7.0.12</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.

**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 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 } ]
```

**Syntax Description**

- **inband** (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 *shared management interface*.

- **out-of-band** (Optional) Displays the out-of-band interface configurations. Out-of-band interfaces are defined by the network operator to specifically receive network management traffic.

- **interface** (Optional) Displays all the protocols that are allowed in the specified interface.

- **type** Interface type. For more information, use the question mark (?) online help function.

- **interface-path-id** Interface instance. Number range varies depending on interface type.

  **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.

- **vrf** (Optional) Displays the Virtual Private Network (VPN) routing and forwarding reference of an out-of-band interface.

**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 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The `vrf` keyword is valid only for out-of-band VRF configurations.

**Task ID**

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

**Examples**

The following sample output displays all the interfaces that are configured as inband or out-of-band interfaces under MPP:

```
RP/0/RP0/CP00: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
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 7.0.12</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.

**Task ID**
- Task ID: Operations
- ID: system  read

**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
```
Public Key Infrastructure Commands

This module describes the commands used to configure Public Key Infrastructure (PKI).

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 8000 Series Routers.

- ca-keypair, on page 167
- clear crypto ca certificates, on page 168
- clear crypto ca crl, on page 169
- crl optional (trustpoint), on page 170
- crypto ca authenticate, on page 171
- crypto ca cancel-enroll, on page 173
- crypto ca enroll, on page 174
- crypto ca import, on page 176
- crypto ca trustpoint, on page 177
- crypto key generate dsa, on page 179
- crypto key generate ecdsa, on page 181
- crypto key generate rsa, on page 183
- crypto key import authentication rsa, on page 185
- crypto key zeroize dsa, on page 186
- crypto key zeroize ecdsa, on page 187
- crypto key zeroize rsa, on page 188
- description (trustpoint), on page 189
- enrollment retry count, on page 190
- enrollment retry period, on page 191
- enrollment terminal, on page 192
- enrollment url, on page 193
- ip-address (trustpoint), on page 195
- key-usage, on page 196
- keypair, on page 198
- lifetime (trustpoint), on page 199
- message-digest, on page 200
- query url, on page 201
- rsakeypair, on page 202
- serial-number (trustpoint), on page 203
• sftp-password (trustpoint), on page 204
• sftp-username (trustpoint), on page 205
• show crypto ca certificates, on page 206
• show crypto ca crls, on page 208
• show crypto ca trustpool policy, on page 209
• show crypto key mypubkey dsa, on page 210
• show crypto key mypubkey ecdsa, on page 211
• show crypto key mypubkey rsa, on page 212
• show platform security integrity dossier, on page 213
• subject-name (trustpoint), on page 215
• utility sign, on page 216
ca-keypair

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>7.0.12</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 root certificate on the router:

```
Router(config-trustp) #ca-keypair rsa system-root-key
Router(config-trustp) #commit
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>keypair, on page 198</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.

`clear crypto ca certificates trustpoint`

**Syntax Description**

- **trustpoint** Trustpoint name.

**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 7.0.12</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 7.0.12</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
```

System Security Command Reference for Cisco 8000 Series Routers
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 7.0.12</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</th>
<th>Operations ID</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}
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ca-name</code></td>
<td>Name of the CA Server.</td>
</tr>
<tr>
<td><code>system-trustpoint</code></td>
<td>Generates self-signed root certificate.</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>7.0.12</td>
<td>This command was introduced.</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.

**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 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 7.0.12</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 202 command in trustpoint configuration mode. If no rsakeypair 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>7.0.12</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 202 command in trustpoint configuration mode. If no `rsakeypair` 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**

<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 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 your 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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name</code></td>
<td>Name of the certification authority (CA). This name is the same name used when the CA was declared with the <code>crypto ca trustpoint</code>, on page 177 command.</td>
</tr>
<tr>
<td><code>certificate</code></td>
<td>None</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 7.0.12</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 Operations 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}
```

**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>7.0.12</td>
<td>This command was introduced.</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 167</td>
<td>Creates the key pair for the root certificate on the router.</td>
</tr>
<tr>
<td>crl optional (trustpoint), on page 170</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 190</td>
<td>Specifies how many times a router resends a certificate request.</td>
</tr>
<tr>
<td>enrollment retry period, on page 191</td>
<td>Specifies the wait period between certificate request retries.</td>
</tr>
<tr>
<td>enrollment terminal, on page 192</td>
<td>Specifies manual cut-and-paste certificate enrollment.</td>
</tr>
<tr>
<td>enrollment url, on page 193</td>
<td>Specifies the URL of the CA.</td>
</tr>
<tr>
<td>ip-address (trustpoint), on page 195</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 196</td>
<td>Specifies the key usage field for the self-enrollment certificate.</td>
</tr>
<tr>
<td>keypair, on page 198</td>
<td>Creates the key pair for the leaf certificate on the router.</td>
</tr>
<tr>
<td>lifetime (trustpoint), on page 199</td>
<td>Configures the lifetime for self-enrollment of certificates.</td>
</tr>
<tr>
<td>message-digest, on page 200</td>
<td>Configures the message digest hashing algorithm for the certificates.</td>
</tr>
<tr>
<td>query url, on page 201</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 202</td>
<td>Specifies a named RSA key pair for this trustpoint.</td>
</tr>
<tr>
<td>serial-number (trustpoint), on page 203</td>
<td>Specifies a router serial number in the certificate request.</td>
</tr>
<tr>
<td>sftp-password (trustpoint), on page 204</td>
<td>Secures the FTP password.</td>
</tr>
<tr>
<td>sftp-username (trustpoint), on page 205</td>
<td>Secures the FTP username.</td>
</tr>
<tr>
<td>subject-name (trustpoint), on page 215</td>
<td>Specifies a subject name in the certificate request.</td>
</tr>
</tbody>
</table>
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 [{system-enroll-key | system-root-key}]
```

**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 7.0.12</td>
<td>This command was introduced.</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.

**Examples**

The following example shows how to generate a 512-bit DSA key:

```
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:
crypto key generate dsa

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 \texttt{crypto key generate ecdsa} command.

\texttt{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>\texttt{nistp256}</td>
<td>Generates an ECDSA key of curve type nistp256, with key size 256 bits.</td>
</tr>
<tr>
<td>\texttt{nistp384}</td>
<td>Generates an ECDSA key of curve type nistp384, with key size 384 bits.</td>
</tr>
<tr>
<td>\texttt{nistp521}</td>
<td>Generates an ECDSA key of curve type nistp521, with key size 521 bits.</td>
</tr>
<tr>
<td>\texttt{system-enroll-key}</td>
<td>Specifies key pair generation for the leaf certificate.</td>
</tr>
<tr>
<td>\texttt{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>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

To remove an ECDSA key, use the \texttt{crypto key zeroize ecdsa} command.

**Example**

The following example shows how to generate an ECDSA key pair:

```
Router# \texttt{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# \texttt{crypto key generate ecdsa system-root-key}
```

This example shows how to generate a ECDSA key pair for the leaf certificate:
crypto key generate ecdsa

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**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usage-keys</td>
<td>(Optional) Generates separate RSA key pairs for signing and encryption.</td>
</tr>
<tr>
<td>general-keys</td>
<td>(Optional) Generates a general-purpose RSA key pair for signing and encryption.</td>
</tr>
<tr>
<td>keypair-label</td>
<td>(Optional) RSA key pair label that names the RSA key pairs.</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**

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></td>
<td></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.

```
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 7.0.12</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 base64 encoded text.

3. Decode the base64 encoded text, and use the for authentication.

**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.

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 7.0.12</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.

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 ecdsa

To delete the Elliptic Curve Digital Signature Algorithm (ECDSA) key pair from your router, use the `crypto key zeroize ecdsa` command.

```
crypto key zeroize ecdsa  [ nistp256 | nistp384 | nistp521 ]
```

**Syntax Description**

- **nistp256**: Deletes an ECDSA key of curve type nistp256, with key size 256 bits.
- **nistp384**: Deletes an ECDSA key of curve type nistp384, with key size 384 bits.
- **nistp521**: Deletes an ECDSA key of curve type nistp521, with key size 521 bits.

**Command Default**

None

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

None

**Task ID**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>execute</td>
</tr>
</tbody>
</table>

**Example**

The following example shows how to delete ECDSA keys from your router:

```
RP/0/RP0/CPU0:router# crypto key zeroize ecdsa nistp384

% Keys to be removed are named the_default
Do you really want to remove these keys ?? [yes/no]: yes
```
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 7.0.12</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 174 command CA.
- Manually remove the certificates from the configuration using the `clear crypto ca certificates` command.

**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 7.0.12</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.

**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>7.0.12</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**

```
read, write
```

**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(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 7.0.12</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 7.0.12</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 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 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/pkiclient.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 7.0.12</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>
<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</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.
### Examples

The following example shows the absolute minimum configuration required to declare a CA:

```bash
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></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.

**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}}
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ca-certificate</code></td>
<td>Specifies the key usage field for the CA certificate.</td>
</tr>
<tr>
<td><code>certificate</code></td>
<td>Specifies the key usage field for the leaf certificate.</td>
</tr>
<tr>
<td><code>crlsign</code></td>
<td>Asserts <code>cRLSign</code> (bit 6) for the key usage field to verify signatures on certificate revocation list (CRL).</td>
</tr>
<tr>
<td><code>digitalsignature</code></td>
<td>Asserts <code>digitalSignature</code> (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).</td>
</tr>
<tr>
<td><code>keycertsign</code></td>
<td>Asserts <code>keyCertSign</code> (bit 5) for the key usage field when the subject public key is used for verifying a signature on public key certificates.</td>
</tr>
<tr>
<td><code>nonrepudiation</code></td>
<td>Asserts <code>nonRepudiation</code> (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.</td>
</tr>
<tr>
<td><code>dataencipherment</code></td>
<td>Asserts <code>dataEncipherment</code> (bit 3) for the key usage field when the subject public key is used for enciphering user data, other than cryptographic keys.</td>
</tr>
<tr>
<td><code>keyagreement</code></td>
<td>Asserts <code>keyAgreement</code> (bit 4) for the key usage field when the subject public key is used for key agreement.</td>
</tr>
<tr>
<td><code>keyencipherment</code></td>
<td>Asserts <code>keyEncipherment</code> (bit 2) for the key usage field when the subject public key is used for key transport.</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>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.
This example shows how to specify the key usage field for the self-enrollment certificate:

```
Router#configure
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>7.0.12</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 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>ca-keypair, on page 167</td>
<td>Creates the key pair for the root certificate on the router.</td>
</tr>
</tbody>
</table>
lifetime (trustpoint)

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>7.0.12</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 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>7.0.12</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
  - **Operations**: read, write

**Examples**

This example shows how to specify SHA256 as the message digest hashing algorithm for the certificate:

```
Router# configure
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 7.0.12</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.

**Examples**

The following example shows the configuration required to declare a CA when the CA supports LDAP:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0/CPU0:router(config-trustp)# query url ldap://my-ldap.domain.com
```
rsakeypair

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 7.0.12</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.

```plaintext
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>7.0.12</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.

### Task ID

<table>
<thead>
<tr>
<th>Task</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 omit a serial number from the root certificate request:

```plaintext
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
```
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**
- `clear text`: Clear text password and is encrypted only for display purposes.
- `password encrypted string`: Enters the password in an encrypted form.

**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 7.0.12</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**
- `crypto`  
- `read, write`

**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**

- `username`: Name of the user.

**Command Default**

None

**Command Modes**

Trustpoint configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</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.

**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 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
```
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.

`show crypto ca certificates`

**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 7.0.12</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).

**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 certificates` command:

```
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:B8:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
   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 7.0.12</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 sample output is from the `show crypto ca crls` command:

```
RP/0/RP0/CPU0:router: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 7.0.12</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 7.0.12</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></td>
<td>crypto</td>
</tr>
<tr>
<td></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 81A0605 2B0E0302 0C3081A2 02022000 024100C8 A36B6179 56B8D620 1F77595C 32F33004 577A9F79 0AA8BDA4 89FB969D 35CD4FCE 5491ED4E 120C657C 610576E5 841696B6 094884FC C92F56E5 B49214E8 70FC4902 1500AB61 5CD63D3 EB0B2BB9 F16030C5 AA0B5D1A DFE50240 73F661EA 9F5799E7 B413BC4 9047B4F2 10A1CFCB 14D989B57 3E0BBA97 9B5120AD F52BDDC7 15B63454 8CB54885 92B639DF 7DC27768 FD269445 202494E 586E81A0 3430002 4071B49E F80F9E4B AF2B62E7 AA817460 87FDD503 C668AD8C D06050B 225CC277 7C0A0974 8072D7D7 2ADDDE42 329FE896 AB01561D 3A414254 6935FDCA 0043BA4F 66
```
show crypto key mypubkey ecdsa

To display the Elliptic Curve Digital Signature Algorithm (ECDSA) public keys for your router, use the `show crypto key mypubkey ecdsa` command.

`show crypto key mypubkey ecdsa` command.

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

**Command Default**
None

**Command Modes**
EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

<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**

```
RP/0/RSP0/CPU0:Router# show crypto key mypubkey ecdsa

Key label: the_default
Type : ECDSA General Curve Nistp256
Degree : 256
Created : 19:10:54 IST Mon Aug 21 2017
Data :
04255331 89B3CC40 BCD5A5A3 3BCCE7FF 522BF88D F3CC300D CEC9D7FD 98796ABB
6A69523F E5FBAB66 804A05BF ECCDABC6 63F73AE8 E89827DD 18EB106A 7735C34A
```
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.

```
show crypto key mypubkey rsa
```

**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 7.0.12</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 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 00CF8CDF
5BFC4065 DA4D164D 76EDB78B 926B1DDE 0383027F BA71BCC6 9D559C4 5BA8670E
35CD19B7 1C973A46 62CC5F8C 82BD596C F292410F 8E83B753 4B71B4AC 41A6B60
F34A2499 EDE11639 F88B4210 B2ACCF5F DD678C36 0D8B7DE1 A2AB5122 9ED947D5
76CF5BCD D9A2039F D02841B0 7F8BFF97 C080B791 10A9ED41 00FB6F40 95020301

Key label: the_default
Type : RSA General purpose
Size : 512
Created : 07:46:15 UTC Fri Mar 17 2006
Data :
305C300D 06092A86 484F8700 0D010105 00034B00 30480241 00C7DE73 7B3EA447
CCE8F3DF DD1327DB 1C30C45 2EEB4981 B1B4B2B2 1AF14665 178058FB 8F6BB6BB
E08C6163 FA08E356 395C8E5F 2AC59383 0706BDDF EC8E5822 9B020301 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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>packages</td>
<td>Displays active package(s) installed.</td>
</tr>
<tr>
<td>reboot-history</td>
<td>Displays reboot history of the node.</td>
</tr>
<tr>
<td>rollback-history</td>
<td>Displays rollback history of the node.</td>
</tr>
<tr>
<td>running-config</td>
<td>Displays the currently committed running configuration on the node, as displayed by <code>show running configuration</code> command.</td>
</tr>
<tr>
<td>system-integrity-snapshot</td>
<td>Displays the system integrity snapshot.</td>
</tr>
<tr>
<td>system-inventory</td>
<td>Displays the system inventory.</td>
</tr>
<tr>
<td>nonce</td>
<td>Specifies the nonce to generate the signature.</td>
</tr>
<tr>
<td>nonce-value</td>
<td>Specifies the nonce value in hexadecimal string format.</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>7.0.12</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:

```plaintext
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
```
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 7.0.12</td>
<td>This command was introduced.</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,OU=ASR
Router(config-trustp)# commit
```
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>7.0.12</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 and Secure Sockets Layer Commands

This module describes the Cisco IOS XR software commands used to configure Secure Shell (SSH) and Secure Socket Layer (SSL).

For detailed information about SSH and SSL concepts, configuration tasks, and examples, see the Implementing Secure Shell chapter in the module in the System Security Configuration Guide for Cisco 8000 Series Routers.

- clear netconf-yang agent session, on page 219
- clear ssh, on page 220
- netconf-yang agent ssh , on page 222
- sftp, on page 223
- sftp (Interactive Mode), on page 226
- show netconf-yang clients, on page 229
- show netconf-yang statistics, on page 230
- show ssh, on page 232
- show ssh history, on page 234
- show ssh history details, on page 236
- show ssh rekey, on page 238
- show ssh session details, on page 239
- show ssl, on page 241
- show tech-support ssh, on page 243
- ssh algorithms cipher, on page 245
- ssh client enable cipher , on page 246
- ssh client knownhost, on page 248
- ssh client source-interface, on page 249
- ssh client vrf, on page 251
- ssh server disable hmac, on page 252
- ssh, on page 253
- ssh server, on page 255
- ssh server algorithms host-key, on page 256
- ssh server enable cipher, on page 257
- ssh server logging, on page 258
- ssh server netconf , on page 259
- ssh server netconf port, on page 260
• ssh server rate-limit, on page 261
• ssh server rekey-time, on page 262
• ssh server rekey-volume, on page 263
• ssh server session-limit, on page 264
• ssh server v2, on page 265
• ssh timeout, on page 266
clear netconf-yang agent session

To clear the specified netconf agent session, use the clear netconf-yang agent session in EXEC mode.

```
clear netconf-yang agent session  session-id
```

**Syntax Description**

*session-id*  The session-id which needs to be cleared.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

No specific guidelines impact the use of this command.

The show netconf-yang clients command can be used to get the required session-id(s).

**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 clear netconf-yang agent session command:

```plaintext
RP/0/RP0/CPU0:router (config) # clear netconf-yang agent session 32125
```
clear ssh

To terminate an incoming or outgoing Secure (SSH) connection, use the clear ssh command.

```
clear ssh {session-id | outgoing session-id}
```

**Syntax Description**

- `session-id` Session ID number of an incoming connection as displayed in the show ssh command output. Range is from 1 to 4294967295.
- `outgoing 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.

**Command Default**

None

**Command Modes**

XR EXEC mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**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

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
```
Secure Shell and Secure Sockets Layer Commands

`clear ssh 0`

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>0/33/1</td>
<td>SESSION_OPEN</td>
<td>cisco</td>
<td>3333::50</td>
<td>v2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Incoming sessions

```
<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>1</td>
<td>0/33/1</td>
<td>SESSION_OPEN</td>
<td>cisco</td>
<td>172.19.72.182</td>
<td>v2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0/33/1</td>
<td>SESSION_OPEN</td>
<td>cisco</td>
<td>3333::50</td>
<td>v2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
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.

```bash
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 7.0.12</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
```
**sftp**

To start the secure FTP (SFTP) client, use the `sftp` command.

```
sftp [username @ host : remote-filenam e] source-filenam e dest-filenam e [source-interface type interface-path-id] [vrf vrf-name]
```

<table>
<thead>
<tr>
<th>Syntax Description</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-filenam e</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-filenam e</code></td>
<td>SFTP source, including the path.</td>
</tr>
<tr>
<td><code>dest-filenam e</code></td>
<td>SFTP destination, including the path.</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>
</tbody>
</table>

**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>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**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.

<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
```

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:
/disk0:/V6copy
    Transferred 308413 Bytes
    308413 bytes copied in 0 sec (421329)bytes/sec
```

```
RP/0/RP0/CPU0:router# dir disk0a:/v6back
```
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
```

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
```

sftp (Interactive Mode)

To enable users to start the secure FTP (SFTP) client, use the `sftp` command.

```
sftp [username @ host : remote-filenam e] [source-interface type interface-path-id] [vrf vrf-name]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Argument</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-filenam e</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>7.0.12</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:

- `by`e
- `cd <path>`
• chmod <mode> <path>
• exit
• get <remote-path> [local-path]
• help
• ls [-alt] [path]
• mkdir <path>
• put <local-path> [remote-path]
• pwd
• quit
• rename <old-path> <new-path>
• rmdir <path>
• rm <path>

The following commands are not supported:
• lcd, lls, lpwd, lumask, lmkdir
• ln, symlink
• chgrp, chown
• !, !command
• ?
• mget, mput

<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 admin is downloading and uploading a file from/to an external SFTP server using an IPv6 address:

RP/0/RP0/CPU0:router#sftp admin@[2:2:2::2]
Connecting to 2:2:2:2...
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>
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>

sftp> put /disk0:/frmRouterdownoad againtoServer
/disk0:/frmRouterdownoad
Transferred 1578 Bytes
1578 bytes copied in 0 sec (14747)bytes/sec
sftp>
show netconf-yang clients

To display the client details for netconf-yang, use the `show netconf-yang clients` command in EXEC mode.

```
show netconf-yang clients
```

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</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></td>
<td>read</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to use the `show netconf-yang clients` command:

```
RP/0/RP0/CPU0:router (config) # sh netconf-yang clients
Netconf clients
client session ID| NC version| client connect time| last OP time| last OP type| <lock>|
22969| 1.1| 0d 0h 0m 2s| 11:11:24| close-session| No|
15389| 1.1| 0d 0h 0m 1s| 11:11:25| get-config| No|
```

**Table 12: Field descriptions**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client session ID</td>
<td>Assigned session identifier</td>
</tr>
<tr>
<td>NC version</td>
<td>Version of the Netconf client as advertised in the hello message</td>
</tr>
<tr>
<td>Client connection time</td>
<td>Time elapsed since the client was connected</td>
</tr>
<tr>
<td>Last OP time</td>
<td>Last operation time</td>
</tr>
<tr>
<td>Last OP type</td>
<td>Last operation type</td>
</tr>
<tr>
<td>Lock (yes or no)</td>
<td>To check if the session holds a lock on the configuration datastore</td>
</tr>
</tbody>
</table>
show netconf-yang statistics

To display the statistical details for netconf-yang, use the **show netconf-yang statistics** command in EXEC mode.

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

None

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</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>config-services</td>
<td>read</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to use the **show netconf-yang statistics** command:

```
RP/0/RP0/CPU0:router (config) # sh netconf-yang statistics
Summary statistics

<table>
<thead>
<tr>
<th>time per request</th>
<th># requests</th>
<th>total time</th>
<th>min time per request</th>
<th>max time per request</th>
</tr>
</thead>
<tbody>
<tr>
<td>other</td>
<td>0</td>
<td>Oh 0m 0s 0ms</td>
<td>Oh 0m 0s 0ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>close-session</td>
<td>0</td>
<td>Oh 0m 0s 3ms</td>
<td>Oh 0m 0s 0ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kill-session</td>
<td>0</td>
<td>Oh 0m 0s 0ms</td>
<td>Oh 0m 0s 0ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>get-schema</td>
<td>0</td>
<td>Oh 0m 0s 0ms</td>
<td>Oh 0m 0s 0ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>get</td>
<td>0</td>
<td>Oh 0m 0s 0ms</td>
<td>Oh 0m 0s 0ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>get-config</td>
<td>1</td>
<td>Oh 0m 0s 1ms</td>
<td>Oh 0m 0s 1ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edit-config</td>
<td>3</td>
<td>Oh 0m 0s 2ms</td>
<td>Oh 0m 0s 0ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>commit</td>
<td>0</td>
<td>Oh 0m 0s 0ms</td>
<td>Oh 0m 0s 0ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cancel-commit</td>
<td>0</td>
<td>Oh 0m 0s 0ms</td>
<td>Oh 0m 0s 0ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lock</td>
<td>0</td>
<td>Oh 0m 0s 0ms</td>
<td>Oh 0m 0s 0ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unlock</td>
<td>0</td>
<td>Oh 0m 0s 0ms</td>
<td>Oh 0m 0s 0ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>discard-changes</td>
<td>0</td>
<td>Oh 0m 0s 0ms</td>
<td>Oh 0m 0s 0ms</td>
<td></td>
</tr>
<tr>
<td>0h 0m 0s 0ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requests</td>
<td>Total number of processed requests of a given type</td>
</tr>
<tr>
<td>Total time</td>
<td>Total processing time of all requests of a given type</td>
</tr>
<tr>
<td>Min time per request</td>
<td>Minimum processing time for a request of a given type</td>
</tr>
<tr>
<td>Max time per request</td>
<td>Maximum processing time for a request of a given type</td>
</tr>
<tr>
<td>Avg time per request</td>
<td>Average processing time for a request type</td>
</tr>
</tbody>
</table>
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 7.0.12</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>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>crypto</td>
<td>read</td>
<td></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 14: 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, Sdp, 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 7.0.12</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` command to display the last hundred SSH sessions that were terminated:

```
RP/0/RP0/CPU0:router# show ssh history

SSH version : Cisco-2.0

id chan pty location userid host ver authentication
connection type

Incoming sessions
1 1 XXXXX 0/RP0/CPU0 root 10.105.227.252 v2 password
Netconf-Subsystem
2 1 XXXXX 0/RP0/CPU0 root 10.105.227.252 v2 password
Netconf-Subsystem
3 1 XXXXX 0/RP0/CPU0 root 10.105.227.252 v2 password
Netconf-Subsystem
4 1 XXXXX 0/RP0/CPU0 root 10.105.227.252 v2 password
Netconf-Subsystem
5 1 XXXXX 0/RP0/CPU0 root 10.105.227.252 v2 password
Netconf-Subsystem
6 1 XXXXX 0/RP0/CPU0 root 10.105.227.252 v2 password
Netconf-Subsystem
7 1 XXXXX 0/RP0/CPU0 root 10.105.227.252 v2 password
Netconf-Subsystem
8 1 XXXXX 0/RP0/CPU0 root 10.105.227.252 v2 password
Netconf-Subsystem
```
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.

**show ssh history details**

**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>7.0.12</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
outmac 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 14:44:08 15-02-18 14:44:09

5 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
```
show ssh history details

hmac-sha2-256  15-02-18 14:50:52  15-02-18 14:50:53
9  ecdh-sha2-nistp256  ssh-rsa   aes128-ctr  aes128-ctr  hmac-sha2-256

This table describes the significant fields shown in the display.

**Table 15: 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 rekey

To display session rekey details such as session id, session rekey count, time to rekey, data to rekey, use the show ssh rekey command.

show ssh rekey

Command Default
None

Command Modes
EXEC

Command History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

Usage Guidelines

The ssh rekey data is updated ten times between two consecutive rekeys.

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 ssh rekey command:

```bash
# show ssh rekey
id  RekeyCount  TimeToRekey(min)  VolumeToRekey(MB)
----------------------------------------------
Incoming Session
0   8           59.5             1024.0
```

This table describes the fields shown in the display.

**Table 16: show ssh rekey Field Descriptions**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rekey Count</td>
<td>Number of times the ssh rekey is generated.</td>
</tr>
<tr>
<td>TimeToRekey</td>
<td>Time remaining (in minutes) before the ssh rekey is regenerated based on the value set using the ssh server rekey-time command.</td>
</tr>
<tr>
<td>VolumeToRekey</td>
<td>Volume remaining (in megabytes) before the ssh rekey is regenerated based on the value set using the ssh server rekey-volume command.</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.

```
show ssh session details
```

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>7.0.12</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.

Task ID

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

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>
</tbody>
</table>
### Field | Description
--- | ---
pubkey | Public key algorithm chosen for key exchange.
incipher | Encryption cipher chosen for the Rx traffic.
outcipher | Encryption cipher chosen for the Tx traffic.
inmac | Authentication (message digest) algorithm chosen for the Rx traffic.
outmac | Authentication (message digest) algorithm chosen for the Tx traffic.
show ssl

To display active Secure Socket Layer (SSL) sessions, use the show ssl command.

```
show ssl [process-id]
```

**Syntax Description**

- `process-id` (Optional) Process ID (PID) of the SSL application. The range is from 1 to 1000000000.

**Command Default**

None

**Command Modes**

EXEC

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

To display a specific process, enter the process ID number. To get a specific process ID number, enter `run pid` from the command line or from a shell.

The absence of any argument produces a display that shows all processes that are running SSL.

**Examples**

The following sample output is from the `show ssl` command:

```
RP/0/RP0/CPU0:router# show ssl
PID Method Type Peer Port Cipher-Suite
1261711 sslv3 Server 172.16.0.5 1296 DES-CBC3-SHA
```

This table describes the fields shown in the display.

**Table 18: show ssl Field Descriptions**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>Process ID of the SSL application.</td>
</tr>
<tr>
<td>Method</td>
<td>Protocol version (sslv2, sslv3, sslv23, or tlsv1).</td>
</tr>
<tr>
<td>Type</td>
<td>SSL client or server.</td>
</tr>
<tr>
<td>Peer</td>
<td>IP address of the SSL peer.</td>
</tr>
<tr>
<td>Port</td>
<td>Port number on which the SSL traffic is sent.</td>
</tr>
</tbody>
</table>

### Field | Description
--- | ---
Cipher-Suite | Exact cipher suite chosen for the SSL traffic. The first portion indicates the encryption, the second portion the hash or integrity method. In the sample display, the encryption is Triple DES and the Integrity (message digest algorithm) is SHA.

### Related Commands

| Command | Description |
--- | ---
run pidin | Displays the process ID for all processes that are running. |
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 7.0.12</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 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:torin-sec1#
```

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></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 the router.</td>
</tr>
<tr>
<td>show ssh rekey</td>
<td>Displays session rekey details such as session id, session rekey count, time 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</td>
<td></td>
</tr>
<tr>
<td>show tty trace error all</td>
<td></td>
</tr>
</tbody>
</table>
**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.

```
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  read, write

  This example shows how to enable CTR cipher on the client and CBC cipher on the server:

  ```
  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>ssh client enable cipher</td>
<td>Enables CBC mode ciphers on the SSH client.</td>
</tr>
<tr>
<td>ssh server enable cipher</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>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

The support for CBC ciphers are disabled by default. 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

### Related Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssh algorithms cipher, on page 245</td>
<td>Configures the list of supported SSH algorithms on the client or on the server.</td>
</tr>
<tr>
<td>ssh server enable cipher, on page 257</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.

```
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 7.0.12</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>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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.</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 7.0.12</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.

**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 IP address of the Management Ethernet interface for all outgoing SSH connections:
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
```

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>vrf-name</th>
<th>Specifies the name of the VRF to be used by the SSH client.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Command Default</th>
<th>None</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Command Modes</th>
<th>XR Config mode</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Command History</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Release</strong></td>
</tr>
<tr>
<td>Release 7.0.12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Usage Guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td>An SSH client can have only one VRF.</td>
</tr>
<tr>
<td>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 <code>ssh client knownhost</code>, on page 248 or <code>ssh client source-interface</code>, on page 249.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task ID</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Task ID</strong></td>
</tr>
<tr>
<td>crypto</td>
</tr>
<tr>
<td>read,</td>
</tr>
<tr>
<td>write</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>The following example shows the SSH client being configured to start with the specified VRF:</td>
</tr>
</tbody>
</table>

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh client vrf green
```
**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**

- **hmac-sha1**
  - Disables the SHA-1 HMAC cryptographic algorithm.

- **hmac-sha2-512**
  - Disables the SHA-2 HMAC cryptographic algorithm.

**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>7.0.12</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
```
To start the Secure Shell (SSH) client connection and enable an outbound connection to an SSH server, use the `ssh` command.

```
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**

- `vrf vrf-name`: Specifies the name of the VRF associated with this connection.
- `ipv4-address`: IPv4 address in A:B:C:D format.
- `ipv6-address`: IPv6 address in X:X::X format.
- `hostname`: Hostname of the remote node. If the hostname has both IPv4 and IPv6 addresses, the IPv6 address is used.
- `username user-id` (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.
- `cipher aes {128-cbc | 192-cbc | 256-cbc}` (Optional): Specifies Advanced Encryption Standard (AES) as the cipher for the SSH client connection. If there is no specification of a particular cipher by the administrator, the client proposes 3DES as the default to ensure compatibility.
- `source interface type interface-path-id` (Optional): Specifies the source IP address of a selected interface for all outgoing SSH connections.
- `command command-name` (Optional): Specifies a remote command. Adding this keyword prompts the SSHv2 server to parse and execute the `ssh` command in non-interactive mode instead of initiating the interactive session.

**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 7.0.12</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 249 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 249 or `ssh client knownhost`, on page 248 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 server**

To bring up the Secure Shell (SSH) server and to configure one or more VRFs for its use, use the `ssh server` command. To stop the SSH server from receiving any further connections for the specified VRF, use the `no` form of this command.

```plaintext
ssh server [{vrf vrf-name | v2}]
no ssh server [{vrf vrf-name | v2}]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vrf</td>
<td>Specifies the name of the VRF to be used by the SSH server. The maximum VRF length is 32 characters.</td>
</tr>
<tr>
<td>vrf-name</td>
<td>If no VRF is specified, the default VRF is assumed.</td>
</tr>
<tr>
<td>v2</td>
<td>Forces the SSH server version to be only 2.</td>
</tr>
</tbody>
</table>

**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>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**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.

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 265` 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, the SSH server is brought up to receive connections for VRF “green”:

```plaintext
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh server vrf green
```
**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.

```plaintext
ssh server algorithms host-key {dsa | ecdsa-nistp256 | ecdsa-nistp384 | ecdsa-nistp521 | rsa}
```

**Syntax Description**
- `dsa` - Allows the specified DSA, ECDSA or RSA algorithm on the SSH server.
- `ecdsa-nistp256` - While configuring this, you can specify the algorithms in any order.
- `ecdsa-nistp384`
- `ecdsa-nistp521`
- `rsa`

**Command Default**
None

**Command Modes**
XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**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**

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

This example shows how to select the `ecdsa` algorithm from the list of auto-generated host-key pairs on the SSH server:

```plaintext
Router#ssh server algorithms host-key ecdsa-nistp521
```
**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>7.0.12</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 ID`: 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**
- `ssh algorithms cipher, on page 245`: Configures the list of supported SSH algorithms on the client or on the server.
- `ssh client enable cipher, on page 246`: Enables CBC mode ciphers on the SSH client.
**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>7.0.12</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 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**

<table>
<thead>
<tr>
<th>Syntax Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>port-number</code></td>
<td>(Optional) Port number for the netconf SSH server (default port number is 830).</td>
</tr>
</tbody>
</table>

**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 7.0.12</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
```
ssh server netconf port

To configure a port for the netconf SSH server, use the `ssh server netconf port` command in the global configuration mode. To return to the default port, use the `no` form of the command.

```
ssh server netconf port port number
no ssh server netconf port port number
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>port</code></td>
<td>Port number for the netconf SSH server (default port number is 830).</td>
</tr>
<tr>
<td><code>port-number</code></td>
<td>Port number for the netconf SSH server (default port number is 830).</td>
</tr>
</tbody>
</table>

**Command Default**

The default port number is 830.

**Command Modes**

Global configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

You must configure the `ssh server netconf` command for at least one VRF, in order to configure a netconf port to enable netconf subsystem support.

**Task ID**

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

**Examples**

This example shows how to use the ssh server netconf port command with port 831:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh server netconf port 831
```

**Related Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ssh server netconf</code></td>
<td>Configures the vrf(s), where netconf subsystem requests are to be received.</td>
</tr>
<tr>
<td><code>netconf-yang agent ssh</code></td>
<td>Configures the <code>ssh netconf-yang backend</code> for the netconf subsystem (Required to allow the system to service netconf-yang requests). For more information, see the <code>System Management Command Reference for Cisco 8000 Series Routers</code>.</td>
</tr>
</tbody>
</table>
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
nossh 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**

Release | Modification
---|---
Release 7.0.12  | This command was introduced.

**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.

**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 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 rekey-time**

To configure rekey of the ssh server key based on time. Use the **no** form of this command to remove the rekey interval.

```
ssh server rekey-time  time in minutes
no ssh server rekey-time
```

**Syntax Description**

- **rekey-time time in minutes** Specifies the rekey-time interval in minutes. The range is between 30 to 1440 minutes.

  **Note** If no time interval is specified, the default interval is considered to be 30 minutes.

**Command Default**

None.

**Command Modes**

Global configuration

**Command History**

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

**Task ID**

- **crypto** read, write

**Examples**

In the following example, the SSH server rekey-interval of 450 minutes is used:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh server rekey-time 450
```
ssh server rekey-volume

To configure a volume-based rekey threshold for an SSH session. Use the no form of this command to remove the volume-based rekey threshold.

```
ssh server rekey-volume  data in megabytes
no ssh server rekey-volume
```

**Syntax Description**

- `rekey-volume data in megabytes`
  Specifies the volume-based rekey threshold in megabytes. The range is between 1024 to 4095 megabytes.
  
  **Note** If no volume threshold is specified, the default size is considered to be 1024 MB.

**Command Default**

None.

**Command Modes**

Global configuration

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**Task ID**

- Task ID
  - crypto read, write

**Examples**

In the following example, the SSH server rekey-volume of 2048 minutes is used:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ssh server rekey-volume 2048
```
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**

- `sessions` Number of incoming SSH sessions allowed across the router. The range is from 1 to 110.

**Note**

Although CLI output option has 110, you are recommended to configure `session-limit` not more than 100. High session count may cause resource exhaustion.

**Command Default**

`sessions`: 64 per router

**Command Modes**

XR Config mode

**Command History**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

**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**

- crypto read, write

**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.

```
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 7.0.12</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:

```
RP/0/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router(config)# ssh server v2
```
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 7.0.12</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
```
FIPS Commands

This module describes the commands used in enabling the FIPS mode.

For detailed information about FIPS configuration tasks, and examples, see the Configuring FIPS Mode chapter in the System Security Configuration Guide for Cisco 8000 Series Routers.

- crypto fips-mode, on page 268
crypto fips-mode

To configure FIPS, use the crypto fips-mode command in the global configuration mode. To remove FIPS configuration, use the no form of this command.

```
crypto fips-mode
no crypto fips-mode
```

### 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>7.0.12</td>
<td>This command was introduced.</td>
</tr>
</tbody>
</table>

### Usage Guidelines

**Note**
You must reload the router for this configuration to take effect.

Use the `show logging` command to display the contents of logging buffers. You can use the `show logging | i fips` command to filter FIPS specific logging messages.

### 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 configure FIPS:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto fips-mode
```